Grado en Ingeniería de Sistemas Audiovisuales (2018-2019)

Trabajo Fin de Grado

"Named Entity Recognition y Topic Modeling: metodología y aplicaciones al procesamiento de texto"

María Ibáñez de Opacua Lomoschitz
Tutor: Simón Roca Sotelo
Leganés, 1 de julio de 2019

RESUMEN

El Procesamiento del Lenguaje Natural (*NLP*) es un campo de la computación que busca caracterizar automáticamente textos o discursos hablados a través de la identificación de patrones y ciertas características. Es un campo muy amplio, que agrupa tareas muy diversas: Reconocimiento de Entidades Nombradas (*NER*), modelado de *topics* o temáticas (*TM*), reducción de las palabras a su lexema o identificación de su función gramatical, interpretación de los sentimientos del autor de un texto, conversión de un texto a discurso escrito o viceversa, etc.

La idea de este proyecto es el desarrollo de una herramienta para etiquetado de entidades clave e identificación de la temática en un texto. Se emplea como corpus de documentos los archivos de subtitulado procedentes de la API de RTVE. En primer lugar, se realiza una revisión bibliográfica de la documentación de las tecnologías existentes en este ámbito, junto con la implementación de un sistema conjunto con una etapa de reconocimiento de entidades y otra de modelado de *topics*. Son evaluadas algunas alternativas para cada una de las etapas, de las cuáles finalmente se selecciona una tecnología que se integra en el sistema final (R y Java con Apache OpenNLP para *NER*, Python con NLTK y Gensim para *TM*).

La calidad del sistema conjunto viene condicionada por la calidad de cada parte, que se evalúa por separado. En la parte de *NER*, los errores son cuantificables, y se emplean métricas matemáticas basadas en el caso de error o acierto (*recall*, *precision*, *accuracy*, *specify*, *F1 score*). En la parte de *TM*, no existe un resultado único de solución ideal al que aproximarse, por lo que la evaluación requiere del empleo de herramientas matemáticas de aproximación, y por ello se exploran varias alternativas (perplejidad, coherencia). Se considera que el trabajo ha cumplido sus objetivos por haberse completado las fases de desarrollo y haberse obtenido resultados razonables en las medidas de evaluación, pero asimismo se plantean nuevas líneas abiertas de trabajo, con las que este proyecto podría desarrollarse más, y en el caso ideal, llegar a implementarse en las plataformas de RTVE, de donde se han obtenido los documentos empleados como base de los sistemas.

Palabras clave: *Natural Language Processing, Machine Learning, Named Entity Recognition, Topic Modeling*

DEDICATORIA

Gracias a todas las personas que han hecho posible este trabajo.

Gracias a Dios. Gracias a mi familia y amigos que me han apoyado.

Gracias también a Manuel Gómez Zotano, por la tutorización de mi trabajo durante las prácticas en RTVE y la resolución de todas las cuestiones que surgían.

Y gracias de forma especial a mi tutor, Simón Roca Sotelo, por las innumerables horas dedicadas a ayudarme este proyecto, que de otra forma no hubiese sido posible.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. MOTIVACIÓN Y OBJETIVOS DEL PROYECTO

1.1. Motivación

En un mundo cada vez más globalizado la cantidad de información que generamos crece exponencialmente. En el momento actual se estima en unos 2,5 quintillones de bytes diarios, que sería equivalente al número de neuronas que sumarían 250 millones de cerebros humanos. El internet de las cosas y las redes sociales multiplican la cantidad de información accesible y generada por cada usuario individual, y requieren sistemas de procesamiento y almacenamiento cada vez más potentes. Para manejar estas cantidades masivas de información (Big Data) que se caracterizan por su velocidad, variedad y volumen, los sistemas convencionales no son suficientes.

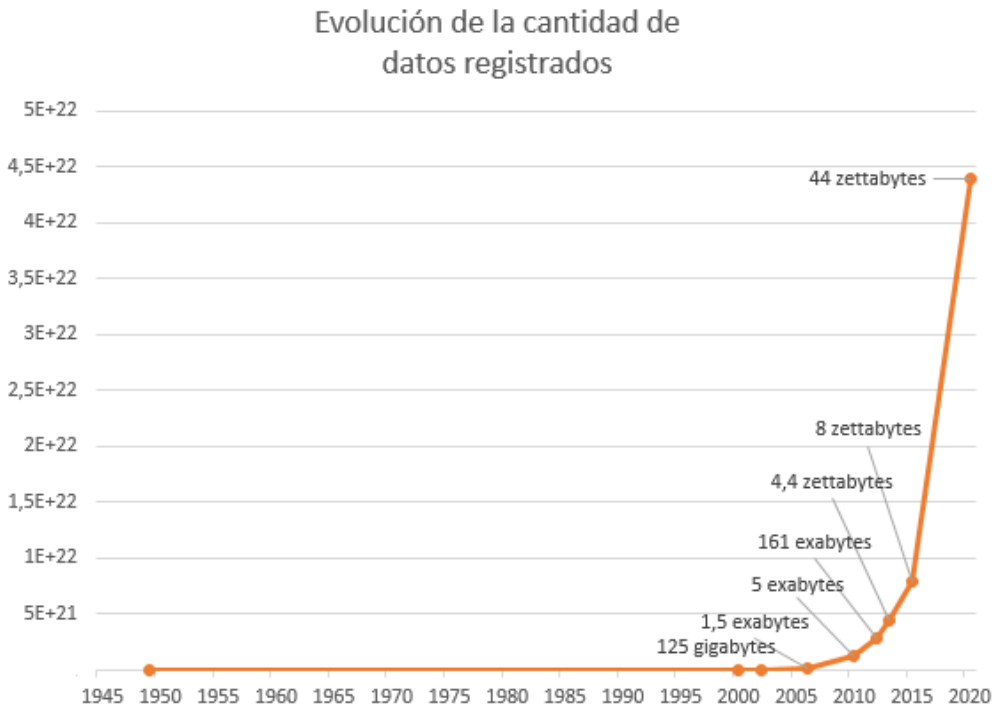


Fig. 1.1: Evolución de la cantidad de datos registrados. [1]Fuente: datos extraídos de informes de la universidad de Berkeley, IDC y la universidad de tecnología de Eindhoven. [1],[2],[3],[4]

En el desarrollo de sistemas de inteligencia artificial para desarrollar sistemas más potentes y eficientes en la resolución de problemas, aparece el aprendizaje automático (*Machine Learning*). Consiste en el desarrollo de sistemas que infieren patrones desconocidos en los datos de entrada, para a través de modelos matemáticos aprender una

[1] 2000: Investigadores de la Universidad de Berkeley cuantifican la cantidad total de información original que cada año se crea en 1,5 exabytes (1,5*1018 bytes, 15.000 millones de GB). [1]
2007: Investigadores del International Data Corporation estiman en 161 exabytes la información creada en 2006, y predicen que la cantidad se sextuplicaría para el 2010, o lo que es lo mismo, se duplicaría cada año y medio, equivalente a 1.610.000 millones de GB. [2] [3]

forma de resolver nuevos casos. Este nuevo enfoque, por la manera en que tiene lugar, se asemeja al aprendizaje experimental de un ser humano. Dentro del *Machine Learning* existen dos casos: aprendizaje supervisado y no supervisado. Para ejemplificar un aprendizaje supervisado, se puede considerar el caso de un niño aprendiendo a caminar, en que un acierto es premiado con celebraciones y un error corregido. Frente a este caso, existe el aprendizaje no supervisado: el sistema es alimentado con numerosos ejemplos, de forma que infiere un patrón o modelo de respuesta que emplea para futuros casos. Un ejemplo es el de un bebé que empieza a hablar, sin recibir ninguna noción de gramática, simplemente por la exposición a ejemplos resueltos. La programación imperativa (reglas) que existía antes de la aparición del aprendizaje automático sería semejante a un código de leyes, en que se detalla minuciosamente cada situación posible y cuál debe ser el comportamiento en este caso.

El lenguaje natural es un tipo de lenguaje originado de forma espontánea entre los seres humanos con el propósito de comunicarse. Es un tipo de discurso escrito o hablado distinto a los lenguajes formales (p.ej., lógica formal, lógica matemática) o los lenguajes de programación. Para analizar automáticamente el lenguaje humano se emplean técnicas de computación, en este ámbito conocidas como *Natural Language Processing* (*NLP*).

La utilidad de estos sistemas reside en la agilización del procesamiento de textos o discursos hablados, la capacidad de extraer parte de información implícita en el lenguaje humano, de la que muchas veces no somos conscientes: distinción de significados de una misma palabra en función del contexto, tema del que se está tratando, referencias a información previamente conocida por ambos interlocutores, juegos de palabras, estado de ánimo de la persona hablando, opinión sobre un tema, etc. Toda esta información implícita en el discurso diferencia una comprensión textual y estricta de lo que se ha dicho exactamente de una comprensión más profunda de lo que se desea decir.

La motivación de este proyecto es realizar un estudio sobre algunas de las herramientas existentes en este campo y desarrollar un sistema capaz de extraer las entidades más relevantes y el tema o los temas de los que se está hablando en un texto. Los documentos serán caracterizados entonces por las palabras clave que aparecen en ellos y por el ámbito común que tienen todos ellos, según unos modelos probabilísticos. De esta forma, el sistema será capaz de devolver los documentos relacionados con una temática o entidad.

1.2. Objetivos

El objetivo fundamental de este proyecto es el estudio de las herramientas existentes en este campo, y el desarrollo de una implementación de un sistema de reconocimiento de entidades nombradas y otro de modelado de *topics*, que puedan después ser relacionados para un conjunto de documentos de distintas temáticas.

Para ello, este trabajo va a desarrollar cinco bloques que constarán de una serie de etapas o hitos intermedios:

1. Documentación y estudio de las tecnologías NLP existentes, comparativas

    a. Búsqueda información, comparativas entre tecnologías
    b. Revisión artículos y documentación específica de cada una
    c. Manejo de la API de RTVE, familiarización con las peticiones HTTP
    d. Desarrollo en Node JS

2. Desarrollo sistema NER

    a. Evaluación sistema R
    b. Evaluación sistema Java
    c. Comparativa sistemas R y Java
    d. Evaluación sistema LUIS (Microsoft Azure)
    e. Evaluación sistema NLU (IBM Watson)
    f. Selección de las mejores herramientas, depurar código

3. Desarrollo sistema TM

    a. Revisión principales métodos de NLP
    b. Familiarización con LDA y sus variables de entrada y salida
    c. Preprocesado específico de los documentos
    d. Obtención de vocabulario
    e. Estudio de tópicos obtenidos
    f. Re-ajuste de parámetros

4. Combinación de las herramientas NER y TM

    a. Sistema conjunto, corregir incompatibilidad
    b. Resultados finales

5. Estudio de los resultados, valoración, documentación del proceso

    a. Medidas de evaluación
    b. Evaluar resultados
    c. Documentar todo el trabajo realizado, memoria
    d. Revisiones

Para estos desarrollos se han empleado sistemas existentes de NLP, que se explican en apartados posteriores, y como base de datos, documentos procedentes de la API de RTVE.

1.3. Estructura de la memoria

Este proyecto consta de 8 capítulos, que a continuación se describen brevemente:

- **Capítulo 1: Motivación y objetivos del proyecto**. Se plantea la situación general del ámbito del que se va a tratar (procesamiento de lenguaje natural), y una visión general de los objetivos del trabajo.

- **Capítulo 2: Planteamiento del problema**. Contexto histórico y actual del NLP, y se estudian los métodos y tecnologías existentes en la actualidad, para tener una visión general del estado del arte actual. Se realiza este desarrollo también en particular para NER y para TM. Asimismo, se plantea el marco legislador, en el que se estudia la legislación existente, y el marco socioeconómico, desglosando los principales intereses de empresas y usuarios de estos sistemas, así como los puntos fuertes y debilidades de las tecnologías existentes.

- **Capítulo 3: Diseño de la solución técnica**. Diagrama de bloques del sistema. Implementación de los sistemas seleccionados, herramientas y datos utilizados para el desarrollo.

- **Capítulo 4: Resultados** de la implementación y evaluación de estos, independientemente para NER y TM, a través de parámetros característicos y medidas de error.

- **Capítulo 5: Organización del proyecto**. Planificación del trabajo, esquema temporal (diagrama de Gantt), presupuesto para el desarrollo del mismo.

- **Capítulo 6: Conclusiones y líneas futuras**. Recapitulación del trabajo realizado, y líneas abiertas de posibles mejoras a partir del proyecto desarrollado.

- **Capítulo 7: Referencias bibliográficas** y documentación empleada para la elaboración de este trabajo.

- **Anexo A: Glosario** de los términos técnicos más importantes a los que se hace referencia en este proyecto.

- **Anexo B: Summary**, resumen en inglés del trabajo realizado en este proyecto.

2. PLANTEAMIENTO DEL PROBLEMA

2.1. Estado del arte

En este apartado se proporciona una visión general del ámbito del Procesamiento del Lenguaje Natural (*NLP*), en el que se encuadra este trabajo.

2.1.1. Introducción histórica

El concepto de **algoritmo** como resolución metódica de problemas de álgebra y cálculo numérico mediante una lista bien definida, ordenada y finita de operaciones es desarrollado por el matemático y astrónomo persa Musa Al-Juarismi (780-850).

La aplicación de algoritmos a la **computación** se da mil años más tarde. En 1777 Charles Mahon, tercer conde de Stanhope inventa la primera máquina lógica, el "demostrador lógico", un aparato de bolsillo capaz de resolver silogismos tradicionales y preguntas elementales de probabilidad. Es en 1822 cuando Charles Babbage desarrolla parcialmente la primera calculadora mecánica capaz de calcular tablas de funciones numéricas por el método de diferencias, y diseña (sin construir) una máquina analítica para ejecutar programas de tabulación o computación, que fue el precedente de los ordenadores actuales. Uniendo este diseño a las tarjetas perforadas[2] y el álgebra de Boole[3], aparece la máquina de Turing[4] y se formaliza el concepto de algoritmo, dando lugar a que en 1938 se desarrolle la primera generación de computadoras (serie Z), durante la Segunda Guerra Mundial.

El problema de la comprensión del **lenguaje** se desarrolla en paralelo. En el siglo XVII, filósofos como Leibniz y Descartes plantearon modelos teóricos de traducción automática mediante códigos que relacionarían palabras en distintos idiomas. Ninguna de ellas fue implementada hasta la década de 1930, con una primera patente para un diccionario bilingüe automático mediante cintas perforadas[5], y otra que incluía un método basado en el esperanto para manejar las funciones gramaticales entre idiomas[6].

En 1950, Turing plantea una prueba para evaluar la "inteligencia" de una máquina, midiendo su habilidad un comportamiento humano: el **test de Turing** [5]. La prueba consiste en una persona que evaluase conversaciones en lenguaje natural entre un humano y una máquina diseñada para generar respuestas similares a las de un humano. Turing sugería que, para pasar la prueba, la máquina debía convencer al evaluador el 70% del tiempo, tras 5 minutos de conversación.

---

[2] 1843: Ada Lovelace propone que se adapten las tarjetas perforadas para permitir que el motor de Babbage repitiese ciertas operaciones.

[3] 1854: George Boole plantea el álgebra de Boole, que reduce a argumentos lógicos las permutaciones de tres operadores básicos: "y", "no", "o" (correspondientes a los símbolos &, ¬, |).

[4] 1938: Alan Turing desarrolla la máquina de Turing

[5] 1933: Georges Artsrouni patenta en Francia un "cerebro mecánico" de propósito general con muchas aplicaciones. http://www.hutchinsweb.me.uk/IJT-2004.pdf, https://www.freecodecamp.org/news/a-history-of-machine-translation-from-the-cold-war-to-deep-learning-f1d335ce8b5/

[6] 1933: Peter Troyanskii patenta en Rusia la "máquina para selección e impresión de palabras al traducir de un idioma a otro"

Aún los sistemas actuales no han conseguido desarrollar una tecnología capaz de superar el test de Turing y convencer a un ser humano de tratarse de una persona, y el análisis del lenguaje natural y toda la información implícita en él continúa en desarrollo.

2.1.2. Inicios del Procesamiento del Lenguaje Natural (*NLP*)

El procesamiento automático del lenguaje natural tiene una finalidad interpretativa, la de obtener información implícita en el contexto o en otros elementos que no aparecen en el discurso. Es un campo de la computación.

Los primeros avances en este campo fueron diseños de traductores automáticos[7]. A partir de 1950 se buscan pautas desde el campo de la lingüística y el estudio de la "gramática generativa", con descripciones basadas en reglas de estructuras sintácticas. Estos sistemas seguían una serie de reglas definidas y estructuradas para afrontar los problemas a resolver.

En dicha década, y hasta 1960, en Estados Unidos se desarrolla un proyecto de Traducción Automática. El Comité Asesor de Procesamiento de Lenguaje Automático (ALPAC, por sus siglas en inglés) redactó un informe sobre los resultados de la financiación y concluyó que "no había habido una traducción automática del texto científico general, ni lo habrá en la perspectiva inmediata", por lo que el proyecto se abandonó. Sin embargo, dieron pie a algunos desarrollos clave:

- **Redes de transición aumentada (*ATN*)**. Software de búsqueda capaz de usar algoritmos gramaticales muy potentes para procesar la sintaxis. Proporcionaron un formalismo para expresar el conocimiento sobre el dominio de la aplicación, en forma de redes de transición extendidas. Del mismo modo, se desarrollaba cómo emplear estas redes para la solución de problemas.

  En el caso de NLP, el conocimiento podría ser sobre la sintaxis en oraciones en inglés, y los problemas, analizar gramaticalmente las oraciones, pero podrían abordarse problemas muy distintos, por ejemplo, planear los movimientos de un robot en un almacén.

- **Caso gramatical**. Estudio de los roles morfológicos y sintácticos de sustantivos, adjetivos o pronombres según la función gramatical que desarrollan. Hay idiomas como el inglés en que la relación entre los verbos y sustantivos se expresa principalmente mediante preposiciones de nexo, mientras que en otros no existen estas preposiciones, sino que la información se encapsula en prefijos, sufijos y declinaciones, o se infiere por el orden en que se enlazan las palabras.

- **Representaciones semánticas**. Aparece la noción de "dependencia conceptual", una forma de expresar el lenguaje a través de unidades semánticas primitivas. Así surgen también los procedimientos semánticos como una representación intermedia entre un sistema de procesamiento de lenguaje y un sistema de bases de datos.

[7] La primera aplicación NLP reconocible fue un diccionario de consulta desarrollado en el Birbeck College (Londres, 1948).

2.2. *NLP*: Métodos que existen actualmente

Como en todos los ámbitos de la computación, al comienzo los procesamientos de lenguaje se desarrollaban partiendo de una lista de instrucciones implementadas manualmente en el código de funcionamiento.

Desde finales de los años 80 y hasta mitad de los 90 se produce la "revolución estadística", que cambia el enfoque que hasta el momento se daba al *NLP* (a través de reglas e instrucciones), para resolverlo también a través de aprendizaje automático (*Machine Learning*).

Las técnicas para *NLP* se clasifican en función de la información que desean caracterizar, como se muestra en la Tabla 2.1.

2.2.1. *Information Retrieval* (*IR*)

Es un proceso de organización de la información (comúnmente, textos) y desarrollo de algoritmos que permitan hacer solicitudes para recuperar los datos de interés. En estos sistemas se fundamentan por ejemplo los buscadores de internet, que, a partir de una consulta, recuperan resultados relacionados con las palabras o condiciones introducidas.

El desarrollo de estas técnicas tiene su origen en el siglo XIX, marcado por estos hitos:

- 1920s - 1930s: Patente de "máquina estadística" que busca patrones y metadata en rollos de documentos microfilmados.

- 1940s - 1950s: Ejército estadounidense trabaja en indexar y recuperar información de investigaciones científicas alemanas durante la Guerra. Preocupación de Estados Unidos por la "brecha científica" frente a la URSS. Se publica "Auto-codificación de documentos para recuperación de información"

- 1960s: Se desarrolla MEDLARS, "sistema de análisis y recuperación de la literatura médica", la primera base de datos importante de lectura mecánica y el sistema de recuperación de lotes.

- 1970s: Primeros sistemas online, desarrollo del hipertexto.

- 1980s: Propuestas de la primera World Wide Web ("www") en el CERN.

- 1990s: Motores de búsqueda implementan funcionalidades antes sólo disponibles en sistemas experimentales de *IR*.

Tabla 2.1:Enumeración de aplicaciones o tareas concretas, clasificadas en función de la información que desean obtener.

Información sintáctica	
Inducción gramatical	Generar un modelo formal de comportamiento sintáctico
Lematización	Eliminación de prefijos o sufijos para obtener la palabra base del diccionario (lema)
Segmentación morfológica	Dividir las palabras en sus morfemas para caracterizarlos
Etiquetado de partes del discurso (Part-of-Speech tagging, *POS*)	Etiquetado de la función de cada palabra en un texto (aun pudiendo desempeñar distintas funciones o ser polisémica)
Análisis gramatical	Obtener el análisis en forma de árbol de la estructura de una oración
Tokenización	Segmentación en palabras y oraciones
Segmentación en oraciones (*SBD*)	Encontrar los límites en que termina cada oración
Stemming	Poda, reducción de las palabras a su raíz o lexema
Chunking	Segmentación de un texto en palabras
Extracción de terminología	Extraer automáticamente los términos relevantes de un corpus
Información semántica	
Semántica léxica	Significado individual o aporte de cada palabra en el contexto
Semántica distributiva	Representaciones semánticas a partir de datos
Traducción automática	Traducciones obtenidas computacionalmente
Reconocimiento de entidades nombradas (*Named Entity Recognition*, *NER*)	Identificación de las entidades que aparecen en el texto y caracterización del tipo de entidad
Generación de lenguaje natural	Transformar información desde bases de datos o unidades semánticas a lenguaje comprensible para un ser humano
Comprensión de lenguaje natural	Convierte texto con lenguaje natural en representaciones más formales, como estructuras de lógica de primer orden, para que sean más fáciles de manejar por otros programas
Reconocimiento óptico de caracteres (*Optical Character Recognition*, *OCR*)	Conversión desde una imagen de texto a texto
Respuesta a preguntas	Capacidad de elaboración de respuestas
Reconocimiento de vinculación textual	Identificar si la veracidad de un texto implica la falsedad de otro, si ambos pueden ser verdaderos, o falsos

Extracción de relaciones entre las entidades que aparecen	Obtención de relaciones entre las entidades que aparecen, comportamiento conjunto de varias entidades
Análisis del sentimiento (*Sentiment Analysis*)	Extracción de información subjetiva sobre el autor del texto, especialmente su "polaridad" en determinado ámbito o ámbitos
Segmentación y reconocimiento de topics	Identificar el tema o temas de un texto
Desambiguación del significado de una palabra en un texto	Desambiguación semántica de las palabras o textos que podrían referirse a varios conceptos o temas
Similitud	Comparación de palabras, trozos de texto y documentos para ver la similitud entre ellos
Clasificación de texto	Asignar categorías o etiquetas a un documento completo, o partes de un documento.
Discurso	
Resumen automático del texto	Sistemas que generan un resumen del texto obtenido computacionalmente
Resolución de correferencias	Qué palabras hacen referencia al mismo objeto o entidad
Análisis del discurso	Amplia variedad de tareas: identificar la estructura del discurso, las relaciones entre las oraciones (p.ej. elaboración, explicación, contraste, etc.), reconocer y clasificar los datos de habla en una porción de texto (p.ej. pregunta de sí/no, pregunta de contenido, declaración, afirmación, etc.)
Habla	
Reconocimiento del habla	Conversión a texto de un discurso hablado o grabado
Chunking del discurso	Segmentación del discurso hablado en palabras
Text-to-Speech	Conversión de texto escrito a mensaje hablado, voz sintética
Diálogo	
Comprensión de instrucciones	Respuesta a instrucciones del usuario
Respuesta a preguntas	Elaboración de respuestas a las preguntas, "imaginación". Utilidad: asistentes virtuales (Siri, Cortana, Alexa, GoogleAllo), chatbots
Adaptación al estado de ánimo o tema de conversación	Capacidad para reconducir la conversación o adaptar el tipo de respuesta

En este proyecto, el sistema desarrollado implementa una solución de análisis semántico basada en dos partes (marcadas en negrita en el listado anterior):

1) Reconocimiento de entidades nombradas (*NER*)
2) Segmentación y reconocimiento de temáticas (*TM*)

2.2.2. *Information Extraction* (*IE*)

Es la obtención automática de información estructurada desde datos semi o no estructurados (en texto, audio, imágenes…). El objetivo es permitir a una máquina operar con estos datos, para lo que son etiquetados los datos en función de la categoría y el contexto. Es una funcionalidad que complementa la gestión básica de un texto (transmisión, almacenamiento y visualización).

Sus orígenes se remontan a finales de 1987, cuando se combina *Machine Learning* con *NLP*. En las décadas posteriores se celebran unas conferencias de comprensión de mensajes (*Message Understanding Conferences*) basadas en competencia, con temáticas como operaciones navales, terrorismo en Latinoamérica, o microelectrónica, que impulsan su desarrollo. Actualmente, *IE* permite a través de métodos estadísticos indexar y clasificar colecciones de documentos a gran escala. La *IE* asume una plantilla previa para los documentos de entrada, a través de eventos o entidades, que cada documento particular tendrá rellenos con sus datos. Es un punto intermedio entre *IR* (*Information Retrieval*, apartado 2.2.1) y las tareas *NLP*. A continuación, se desarrolla una de sus subtareas.

2.2.3. Reconocimiento de entidades nombradas (*NER*)

El reconocimiento de entidades nombradas es una subtarea de *Information Extraction* (obtención de información), que busca localizar y clasificar las apariciones de una entidad nombrada en texto no estructurado en unas categorías predefinidas, como pueden ser los nombres de personas, organizaciones, lugares, códigos médicos, expresiones de tiempo, cantidades, valores monetarios, porcentajes, etc.

Consiste en la identificación de los términos que se refieren a entidades reales y concretas, frente a sustantivos genéricos que no particularizan, como podrían ser “una mujer”, “el campo” o “asociaciones”. Hay muchos tipos de entidades identificables en un texto, y según la tecnología que se emplee y el entrenamiento de los modelos se pueden identificar incluso entidades personalizadas. Otros ejemplos son: fechas, cantidades, nombres de proteínas o moléculas, compañías tecnológicas, números de teléfono, porcentajes, códigos postales…

Para el reconocimiento de las entidades más usuales existen modelos ya entrenados, pero las tecnologías permiten reentrenar modelos para identificar las entidades que nos interesen. Esto es especialmente útil cuando los modelos existentes no tienen soporte para identificar las entidades en cierto idioma.

Existen tres tipos de aprendizaje:

- Modelos basados en reglas (aproximación convencional, instrucciones)
- Modelos basados en etiquetas (aprendizaje supervisado, en que durante el aprendizaje el modelo se reajusta comparando los resultados que obtiene con las soluciones que debería obtener)
- Modelos no supervisados

En este proyecto se emplean técnicas NER con modelos basados en etiquetas (aprendizaje supervisado). Tienen la ventaja de que el mismo procesamiento con que el modelo aprende puede utilizarse igualmente para otros tipos de entidades que se deseen, o para aprender en otros idiomas, frente a los modelos basados en reglas, que requerirían volver a codificar todos los casos para adaptarse a las nuevas estructuras o idiomas. El proceso para **entrenar un modelo supervisado** para reconocer entidades es el siguiente:

1. Se prepara un set de documentos lo más extenso posible, etiquetados correctamente con el tipo de entidades que se desean localizar. El etiquetado de las entidades normalmente sigue el patrón del ejemplo:

   > "El jueves a las <START:time>12h</END> tendrá lugar el último acto oficial del <START:person>rey Juan Carlos</END>, será en el <START:time>monasterio de El Escorial</END>, según ha comunicado recientemente su hijo <START:person>Felipe</END>."

   Por lo general cada modelo se entrena para identificar una sola entidad, por lo que estarían etiquetadas sólo las entidades de un tipo.

2. Este set de documentos se utiliza para **entrenar** el **nuevo modelo** de reconocimiento de entidades. El sistema, de forma opaca para el usuario, aplica funciones matemáticas para inferir un patrón o unas características que el sistema encuentra en común en todas las entidades que desea localizar. Típicamente se sugiere que exista heterogeneidad entre el conjunto de muestras de entrenamiento, para que el sistema sea capaz de gestionar casos variados, así como una cantidad suficiente de muestras. Para un número suficiente de ejemplos este sistema suele proporcionar buenos resultados.

3. Por último, la lista de entidades obtenidas puede visualizarse como una lista, o, según la herramienta, en un entorno más visual, que permita después emplear estas entidades para procesamientos posteriores.

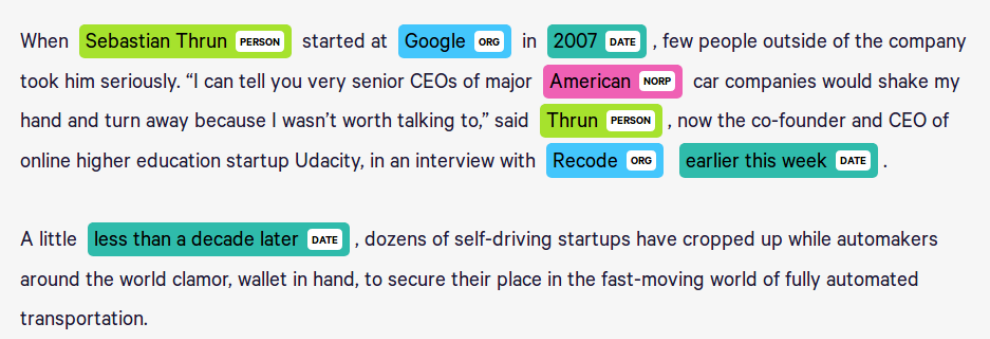


Fig. 2.1: Ejemplo de texto en el que se han reconocido las entidades de persona, fecha, organización, y grupo político/étnico/nacionalidad/religioso. Fuente: Hackernoon. [7]

Por ser un tema en auge existen muchos proyectos y compañías desarrollando soluciones tecnológicas en el ámbito del procesamiento del lenguaje natural, por ejemplo, en departamentos de grandes compañías como Google, Facebook y Microsoft. A continuación, se realiza un estudio comparativo de algunas tecnologías existentes en el mercado (Tabla 2.2).

Tabla 2.2: Principales plataformas NLP con soporte NER y sus características. Fuentes: páginas oficiales de cada tecnología, Wikipedia, artículos comparativos [8], [9], [10]

	Plataformas para NER	Descripción	User Interface	Basada en	Tareas NLP	Soporte adicional	Idiomas disponibles	Licencia, precio	Desarrollada por
1	GATE (General Architecture for Text Engineering)	Admite NER en muchos idiomas y situaciones, con interfaz gráfica y API de Java. Entorno de desarrollo integrado	✓	Java	Sistema de extracción de información, "ANNIE" (A Nearly-New Information Extraction System), módulos: tokenizador, gazetador, divisor de oraciones, etiquetador gramatical (POS), transductor con NER y etiquetador de correferencia.	Puede ejecutarse en cualquier plataforma con soporte Java, sin soporte específico	Inglés, alemán, chino, árabe, búlgaro, hindi, italiano, cebuano, rumano, ruso y danés.	*Open source*, gratis	Universidad de Sheffield (Inglaterra), 1995 hasta la actualidad
2	Apache OpenNLP	Entorno para desarrollo de herramientas NER estadísticas y basadas en reglas.	✗	Java (JDK), requiere Apache Maven	Detección de lenguaje, tokenización, segmentación de oraciones, etiquetado POS, NER, chunking, análisis y resolución de correferencias.	Puede ejecutarse en cualquier plataforma con soporte Java, sin soporte específico. P.ej.: R: openNLP	Modelos preentrenados en inglés, español, alemán, portugués, danés, holandés y sueco	*Open source*, gratis	Apache Software Fundation, 2004 hasta la actualidad
3	Stanford NER (o también CRFClassifier)	Plataforma para NLP estadístico, basado en reglas o aprendizaje profundo (Deep Learning)	✗	Java 1.8+, requiere CoreNLP	Implementación general de modelos secuenciales de Campo Aleatorio Condicional (CRF): entrenando modelos permite implementar NER o realizar otras tareas	Apache Tika, JavaScript/npm, .NET/F#/C# Perl, PHP, Python (NLTK 2.0+), Ruby, UIMA	Inglés, modelos en español, francés, alemán, chino, árabe, y para italiano, portugués, sueco (creados por usuarios)	*Open source*, gratis	Universidad de Stanford (EE.UU.), 2010 hasta la actualidad
4	NLTK	Plataforma para desarrollar NLP en Python	✗	Python	Paquete de Python que proporciona corpora de lenguaje natural (corpus) y API para muchos algoritmos	–	16 (Snowball), + árabe (ISRIStemmer)	*Open source*, gratis	Equipo de desarrollo NLTK, 2005 hasta la actualidad
5	Snowball	Lenguaje para crear stemmers (podadores)	✗	–	Lenguaje para crear sistemas stemming (poda de palabras a su raíz o lexema), compatibilidad con otros lenguajes de programación	El compilador Snowball permite traducirse a ISO C, ANSI C, C#, Go, Java, Javascript, Object Pascal, Python y Rust	Inglés, español, alemán, italiano, francés, húngaro, danés, holandés, finés, portugués, noruego, rumano, ruso, sueco	*Open source*, gratis	Creador Dr Martin Porter, contribuciones por usuarios, 2002 hasta la actualidad

	Plataformas para NER	Descripción	User Interface	Basada en	Tareas NLP	Soporte adicional	Idiomas disponibles	Licencia, precio	Desarrollada por
6	Spacy	Librería código abierto para NER estadística y rápida, visualizador open-source de entidades	✗	Python, Cython	Tokenización no destructiva, NER, compatibilidad con "tokenización alfa" (>25 idiomas), modelos estadísticos (8 idiomas), vectores de palabras pre-entrenados, POS tagging, análisis de dependencias etiquetadas, segmentación de oraciones según sintaxis, clasificación de texto, visualizadores incorporados para sintaxis y entidades nombradas, integración de Deep Learning	–	Español, inglés, alemán, portugués, francés, italiano, holandés y NER multi-idioma, tokenización para más idiomas	*Open source*, gratis	Autor original Matthew Honnibal, desarrollo en Explosion AI y otros, 2015 hasta actualidad
7	Waikato Environment for Knowledge Analysis (WEKA) R -> RWeka TM (paquete Text Mining)	Conjunto de algoritmos para Data Mining (análisis de datos y modelos predictivos)	✗	Java	Algoritmos para Data Mining. NLP: tokenización, stemming. Aprendizaje automático, extracción de datos, preprocesamiento, clasificación, regresión, agrupación, reglas de asociación, selección de atributos, experimentos, flujo de trabajo y visualización.	R -> RWeka	Inglés	Open source, gratis	Universidad Waikato (NZ), 1999 hasta la actualidad; RWeka -> CRAN R project
8	TextBlob	Librería para procesar texto, API para NLP	✗	Python	Segmentación de oraciones, POS tagging, Sentiment Analysis, clasificación (árbol de decisiones, Bayes ingenuo), traducción y detección de idiomas (Google Translate), tokenización, frecuencias de palabras y frases, parsing, n-gramas, inflexión de palabras (pluralización y singularización) y lematización, corrección ortográfica, extensiones para añadir nuevos modelos o idiomas, integración de WordNet (base de datos léxica en inglés)	–	Inglés, alemán, francés	*Open source*, gratis	TextBlob Project, 2013 hasta la actualidad
9	Carrot2	Motor de agrupamiento de resultados de búsqueda de código abierto	✓	Java	Agrupar colecciones de resultados en categorías temáticas, componentes para resultados de búsqueda en varias fuentes	API nativa de C # / .NET. Para otras (PHP, Ruby), a través de la interfaz REST	Inglés, español, alemán, italiano, danés, chino, griego, holandés, árabe, húngaro, italiano… (19)	*Open source*, gratis	Carrot2 project, 2002 hasta la actualidad

	Plataformas para NER	Descripción	User Interface	Basada en	Tareas NLP	Soporte adicional	Idiomas disponibles	Licencia, precio	Desarrollada por
10	Gensim	Librería de código abierto lista para la producción para el modelado de tópicos sin supervisión y el procesamiento del lenguaje natural, utilizando el aprendizaje estadístico moderno	✗	Python, Cython	Herramientas de visualización para modelado de tópicos, implementaciones en paralelo de secuencias de los algoritmos fastText, word2vec y doc2vec, análisis semánticos latentes (LSA, LSI, SVD), factorización matricial no negativa (NMF), asignación latente de Dirichlet (LDA), tf- idf y proyecciones aleatorias	–	Inglés	*Freeware*, código propietario gratuito	ScaleText, Radim Rehurek 2008 hasta la actualidad
11	NER and Disambiguation (NERD)	Plataforma web para extraer y desambiguar entidades	✓	Java, Python, NodeJS, Ruby	API con soporte para NER y NED, plataforma web, cuenta con panel. Proyecto Linked Data vision con reconocimiento de entidades en imágenes y vídeo (NLP Interchange Format, NIF)	Java, Python, NodeJS, Ruby	Inglés	*Open source*, gratis	NERD (Francia), 2011-2012 (última versión)
12	Google Cloud NLP, Natural Language API	API REST para modelizar la estructura y el significado del texto con modelos de aprendizaje automático y de modelos personalizados cen AutoML NaturalLanguage	✓	Proyecto en GCP Console	Modelos personalizados, API REST, modelos AutoML Natural Language, NER, Sentiment Annalysis, clasificador de texto, analizador de entidades…	C#, Go, Java, NodeJS, PHP, Python, Ruby	Combinable con otras APIs de Google: API Google Cloud Speech para interpreter audio, OCR para texto escaneado, Cloud Translation API para traducir a otros idiomas.	Software propietario, precios según plan contratado (en función de la potencia requerida)	Google, 2016 hasta la actualidad

	Plataformas para NER	Descripción	Tareas NLP	Soporte adicional	Licencia, precio	Desarrollada
13	Watson Natural Language Understanding	Servicio cloud a demanda para procesos de NLP	NER, Sentiment Annalysis, metadatos, topics, roles semánticos	Curl, Java, Go, NodeJS, Python, Ruby, Swift. + otros módulos	Software propietario, precios según plan	IBM Watson, 2017-actual.
14	Azure LUIS (Language Understanding Intelligence)	Servicio cloud API para aplicar Deep Learning a NLP	Modelos NER re-entrenables, Speech to Text, análisis de texto	Combinable con otros módulos Cognitive System. Peticiones http de json		Azure Microsoft, 2016-actual.
15	Otras tecnologías:	Rapid Miner Text Miner extension; Coding Annalysis Toolkit (CAT); KH Coder; QDA Miner Lite; VisualText; TAMS; Pattern; Datumbox; Apache Mahout; Textable; Apache UIMA; Apache Stanbol; KNime Text Processing; LingPipe; Gensim; Aika; LPU; Distributed Machine Toolkit; Baleen; Cogcomp-NER; ParalellDots; Minimal NER (MER); Dataturks; BRAT.				

2.2.4. Modelado de tópicos (*TM*)

Como se explicaba en el apartado anterior ("2.2.1 *NER*"), existen tres tipos de aprendizaje:

- Modelos basados en reglas (aproximación convencional, instrucciones)
- Modelos basados en etiquetas (aprendizaje supervisado)
- Modelos no supervisados (aprendizaje no supervisado)

Las técnicas de *Topic Modeling* pertenecen a los modelos no supervisados, puesto que no hay una "plantilla" con las soluciones correctas con las que comparar durante el aprendizaje. El modelo extrae unas conclusiones, y el diseñador las evalúa. Si considera que son acertadas o buenas, mantiene el modelo. Si se considera que pueden mejorarse los resultados, mediante cambios en el sistema implementado se pueden ensayar otras situaciones y evaluar los resultados para cada caso.

En el campo del procesamiento de lenguaje natural, un **modelo de tópicos** es un modelo estadístico que trata de identificar o detectar los patrones implícitos que caracterizan las temáticas o ámbitos en común de las palabras que aparecen en un texto, de forma no supervisada, como se ha explicado. El sistema consta de tres fases:

1. Durante el **entrenamiento** del sistema se le proporciona un **"corpus"** de un gran número de documentos (se espera que el tamaño sea grande pero no hay una regla fija, sino que empíricamente se debe adaptar en función de los resultados obtenidos para el caso en el que se esté trabajando) con alguna relación entre sí, por ejemplo, tener una estructura similar, tratar sobre un mismo tema o ser el mismo tipo de documentos en cuanto a extensión. El sistema procesa los documentos y analiza en qué documentos suelen aparecer las palabras, y en cuáles suelen coocurrir las mismas.

   Los **grupos de palabras** que tienden a explicar documentos similares conformarán un "*topic*" o tema en común. Ese *topic* se caracteriza como una mezcla probabilística de palabras: unos términos aparecerán con muy altas probabilidades dentro del *topic*, mientras que otros tendrán probabilidad casi nula por no ser característicos para ese tema.

   Durante el proceso de entrenamiento, además del corpus de documentos se puede seleccionar el número de *topics* que se desea obtener, así como los parámetros alfa y beta, que condicionan la granularidad y exclusividad de los *topics*. Existen algunos procedimientos automáticos para seleccionar los parámetros óptimos pero que por su complejidad no han sido explorados en este trabajo, y pueden plantearse como líneas abiertas de mejora.

   El modelo creado durante el entrenamiento explica los patrones encontrados en los documentos del corpus, y proporciona una lista de los *topics* aprendidos, así como la distribución de *topics* de los documentos de entrenamiento (probabilidades de cada temática de aparecer dentro de cada documento).

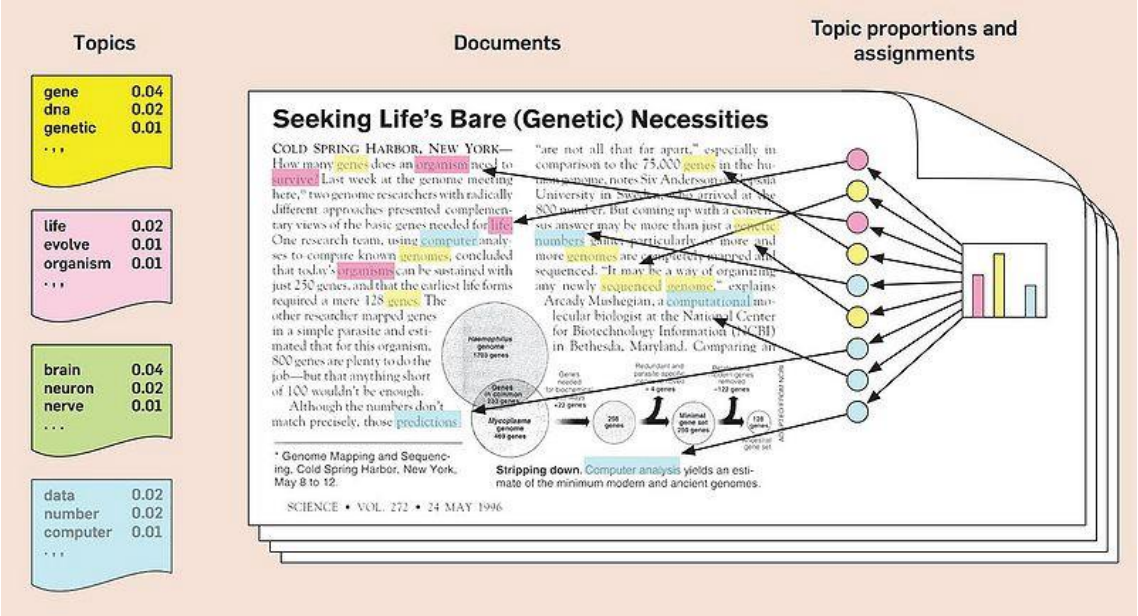


Fig. 2.2: Fundamentos modelos probabilísticos de TM. Fuente: D. M. Blei, 2012 [6]

En el ejemplo de la Fig. 2.2, los *topics* extraídos de procesar el corpus de documentos son los siguientes:

- Gen (4% de probabilidad de aparecer si el documento trata de este *topic*), ADN (2%), genética (1%), etc.
- Vida (2%), evolución (1%), organismo (1%), etc.
- Cerebro (4%), neurona (2%), nervio (1%), etc.
- Datos (2%), número (2%), ordenador (1%), etc.

2. Durante el entrenamiento, el sistema ha caracterizado los documentos del corpus a través de una **lista de temáticas** (conformadas por unos términos y sus probabilidades de aparecer) y la proporción de cada temática en los documentos. Empleando este modelo de tópicos ya creado, se puede obtener una predicción de la composición de tópicos en un documento de test, es decir, un nuevo documento no utilizado durante el entrenamiento.

3. Los resultados obtenidos pueden visualizarse en algún entorno gráfico, o emplearse para otros procesamientos posteriores. El modelo extrae conclusiones en dos aspectos:

   - La **distribución de *topics* en los documentos**: La aparición de los *topics* dentro de cada documento: si el texto pertenece principalmente a uno de los *topics*, si se distribuye en varios *topics* dentro del mismo documento, si los conjuntos de palabras que se consideran un *topic* proceden de un mismo documento, etc.

- La **distribución de los términos característicos dentro de un *topic***: Se modela la aparición de ciertas palabras dentro unos *topics* (proporcionalmente, cuáles se repiten más dentro de un tema, cuáles suelen aparecer conjuntamente, en qué temas suelen aparecer conjuntos comunes, etc.). En el ejemplo de la Fig. 2.2, la palabra "gen" es el doble de probable dentro de un texto del *topic1* (relativo a la genética), que la palabra "ADN".

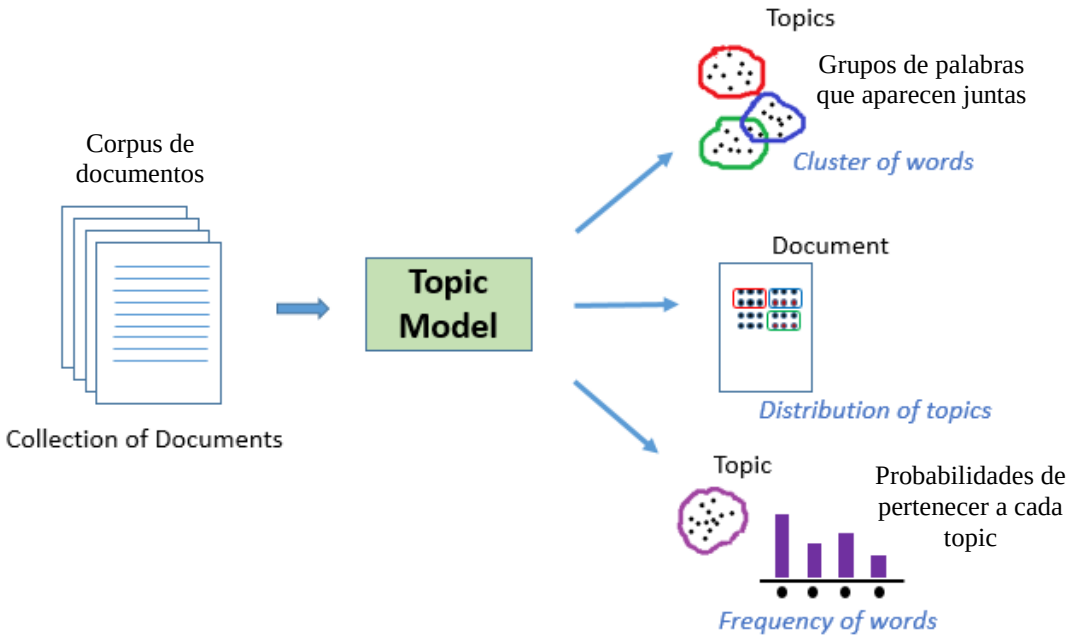


Fig. 2.3. Esquema del funcionamiento completo de un sistema de Topic Modeling. Fuente: Analytics Vidhya [11]

Unidas a las tecnologías presentadas en el apartado anterior, a continuación, se enumeran algunos de los sistemas más relevantes en este campo:

- ***Latent Semantic Indexing*** (*LSI*), en origen llamado *Latent Semantic* Analysis (*LSA*). Es la primera técnica de modelado de *topics* que se desarrolló (1988, [12]), basada en operaciones algebraicas matriciales sobre una matriz de ocurrencias de un término frente a los documentos en que se producen las ocurrencias (Fig. 2.4). Mediante la descomposición de la matriz en valores singulares se busca identificar patrones entre los términos y las temáticas en una colección de textos no-estructurados (Fig. 2.5).

  En la matriz (Fig. 2.4), las filas corresponden a los términos del documento, y las columnas a los documentos que se están analizando. En cada celda se da el cruce entre un documento y un término, y tiene mayor o menor importancia relativa (en la figura, más o menos oscuro) en función de la relación "*tf–idf*" ("*term frequency – inverse document frequency*"). Es un factor muy empleado en sistemas de *Information Retrieval*, que aumenta cuanto más frecuente es un término en un documento, y se compensa disminuyendo conforme aparece en más documentos. Así se compensan las palabras que no aportan información por aparecer muy

frecuentemente en cualquier documento (por ejemplo, adverbios, artículos o preposiciones, o palabras muy genéricas como “cosa”, “algo”).

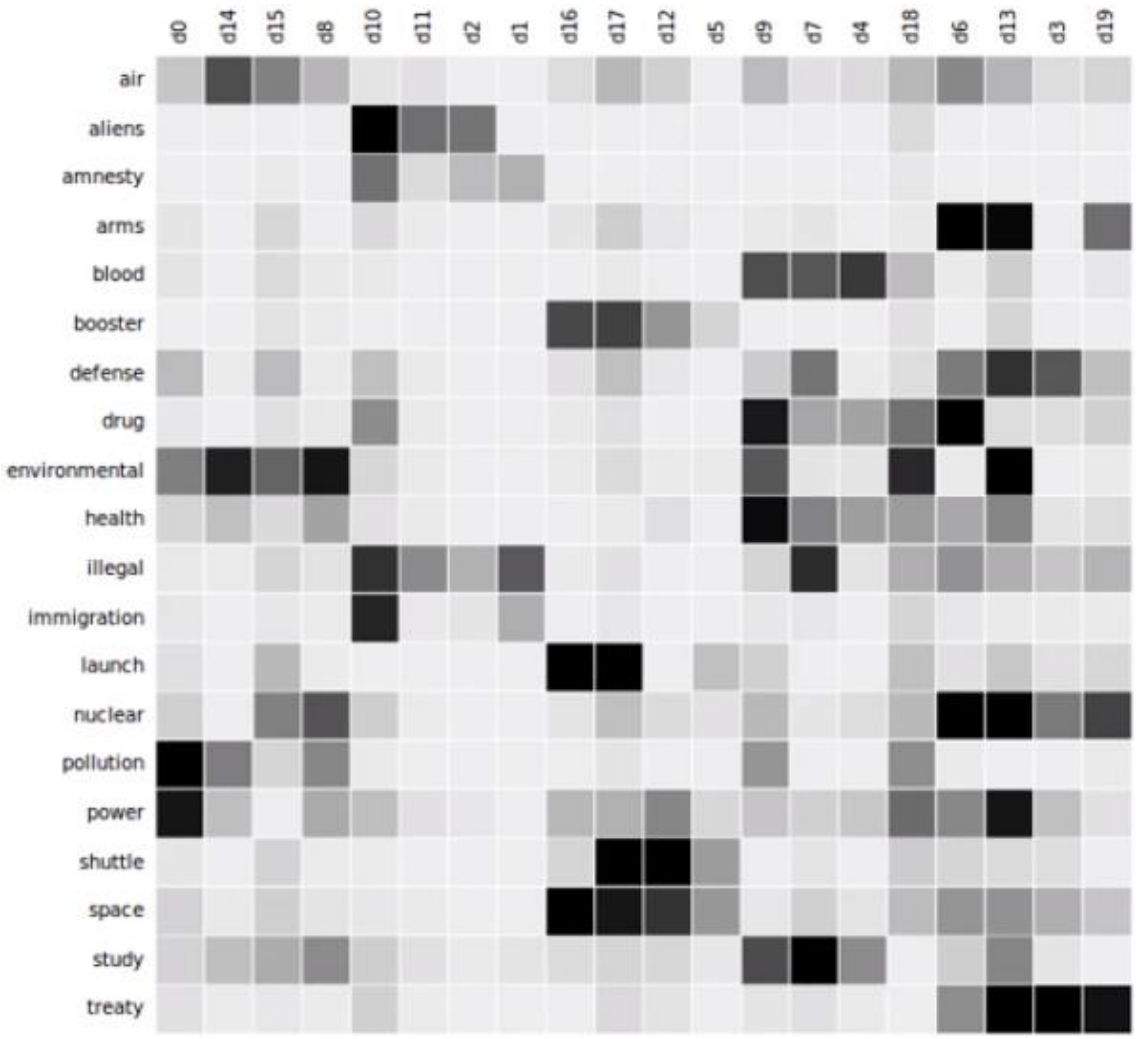


Fig. 2.4: Matriz de ocurrencias para aplicar LSA. En oscuro aparecen los términos (fila) muy importantes en cierto documento (columna) pero que no son muy frecuentes en todos. Aquellos términos que aparecen indistintamente en cualquier documento aparecerán en color claro, porque aportan poca información, igual que los términos que aparecen pocas veces en un documento. Fuente: Wikipedia [13]

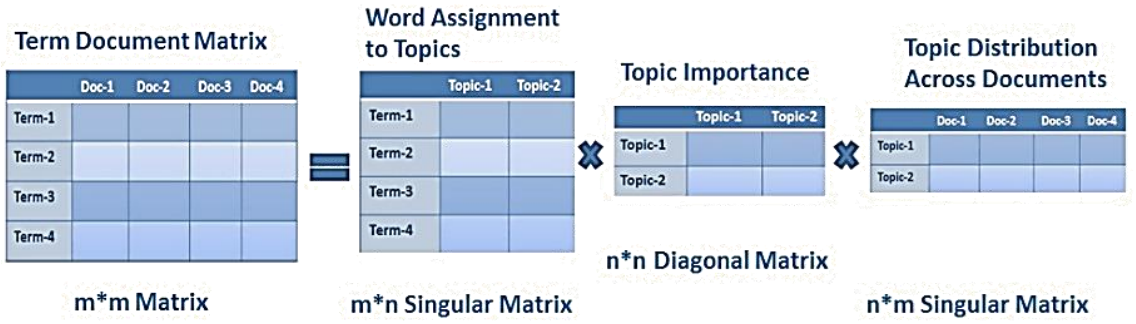


Fig. 2.5: Proceso de factorización en valores singulares (*Singular Value Decomposition*, *SVD*). Fuente: Datacamp [15]

- ***Latent Dirichlet Allocation*** (*LDA*) [12]. Es un modelo de aprendizaje automático posterior, que introduce los modelos bayesianos para obtener las distribuciones estadísticas de los términos en los documentos. Es un modelo generativo estadístico en que el conjunto de observaciones (términos) se comporta según grupos no observados. Siendo términos las observaciones, se considera que los documentos son mezclas de un pequeño número de *topics*, que justifican la aparición de esos términos en un documento. En este modelo, se asume una distribución "Dirichlet" de las categorías, pero existen variantes con otras distribuciones (*probabilistic LDA*, *pLDA*), que, no obstante, no proporcionan tan buenos resultados.

  Basados en el modelo *LDA* existen otras variantes que introducen nuevos desarrollos:

- ***Hierarchical LDA*** ("*hLDA*, *LDA* jerárquico") [17] Se trata de un modelo jerárquico, que antes de aplicar una distribución a los datos calcula una aproximación observando el comportamiento de pequeñas particiones de los datos mediante el "proceso anidado del restaurante chino" ("*nested Chinese restaurant process*"). El proceso asigna a cada muestra individualmente una categoría (en el ejemplo, a cada cliente que llega se le sienta en una nueva mesa o en una ya ocupada, que será más atractiva cuanta más gente tenga). Es una aproximación no-paramétrica, que permite altos factores de ramificación y se puede adaptar a colecciones de datos en crecimiento.

- ***Correlated Topic Modeling*** (*CTM*, modelado de *topics* correlados) [18]. Tomando como base el postulado del *LDA*, que establece que cada documento será una mezcla de distintos *topics*, este modelo trata de encontrar la correlación entre estas categorías de las que se habla en un mismo documento. En el artículo en que se presenta, la aproximación se hace con un modelo de regresión logística.

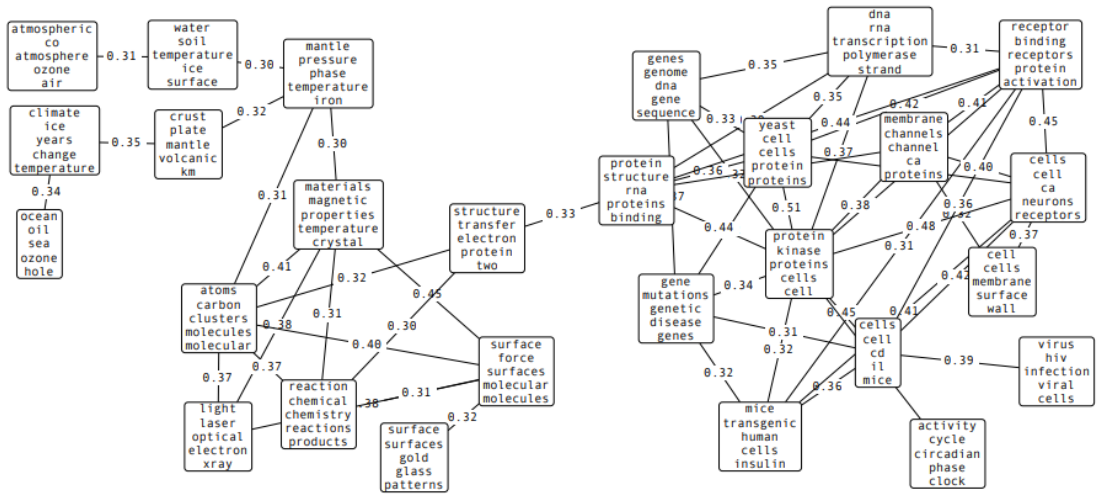


Fig. 2.6: Una parte de la gráfica de *topics* aprendidos de 15,744 artículos de OCR de *Science*. Cada nodo representa un tema y está etiquetado con las cinco palabras más probables de su distribución; y las conexiones con la correlación entre temas. Fuente: Griffiths y Blei [19]

- ***Dynamic Topic Modeling*** (*DTM*) [20]. Son modelos generativos que permiten observar la evolución temporal de los *topics* (no observados) de un conjunto de documentos. Es una extensión de los modelos *LDA* que permite modelar la evolución de los términos dentro de un *topic* a lo largo del tiempo. Por ejemplo, en artículos científicos de 1930 no aparecerán los mismos términos que en artículos científicos de la misma categoría de 1980 o que en 2010. (Figura 2.6)

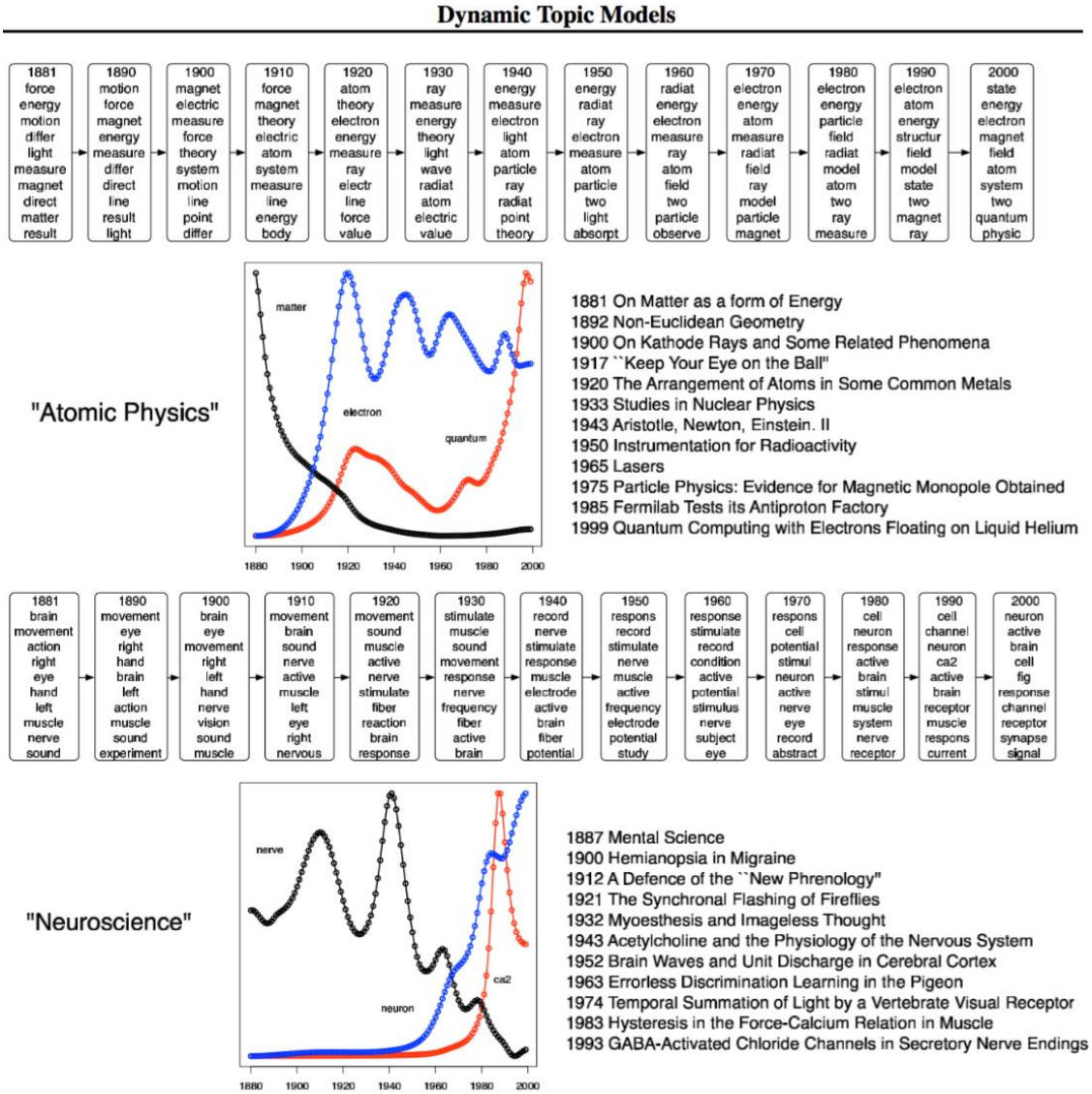


Fig. 2.7: Ejemplo de Dynamic Topic Modeling dentro de los *topics* "física atómica" y "neurociencia". Fuente: Blei [21]

- ***Autoencoding Variational Inference for Topic Modeling*** (*AVI TM*) [23]. El principal problema de los sistemas de *TM* es que para implementar cualquier cambio se necesita desarrollar matemáticamente un nuevo algoritmo de inferencia. Mediante el uso de redes neuronales (y, por tanto, de inteligencia artificial) se pueden obtener modelos re-entrenables, reajustables.

2.3. Marco legislador

Se expone a continuación la legislación aplicable a los datos manejados y el ámbito de este proyecto. Se hace una diferenciación en cuatro ámbitos:

- Legislación para proveedores de contenido audiovisual, aplicable a la corporación RTVE y su creación y difusión de contenidos. (Apartado 2.3.1).
- Tratamiento y almacenamiento de datos sensibles y de carácter personal, que podrían ser aplicables para ciertos contenidos de la API obtenidos, por ejemplo, programas monográficos, de entrevistas o reportajes sobre una persona famosa o conocida, o con un cargo público. (Apartado 2.3.2).
- Leyes de propiedad intelectual. (Apartado 2.3.3).
- Licencias software para las tecnologías empleadas. (Apartado 2.3.4).

2.3.1. Contenidos audiovisuales

- **Código de Derecho Audiovisual.** [24] La normativa se desglosa entre: normativa común a radio y televisión; normativa TV; normativa radio; radio digital; cinematografía. Aunque los contenidos con los que se ha trabajado son documentos escritos, parten del subtitulado de archivos televisivos, por lo que se presta especial atención a la normativa para este ámbito. Se exponen a continuación los principales puntos de la última ley aprobada, de carácter común.

- **Ley 7/2010, de 31 de marzo, General de la Comunicación Audiovisual. «BOE» núm. 79, de 1 de abril de 2010. Referencia: BOE-A-2010-5292** [25], como parte del proyecto de reforma audiovisual emprendido por el Gobierno, complementada con la aprobación de la Ley 17/2006 de la Radio y la Televisión de Titularidad Estatal y complementada con la Ley de Financiación de la Corporación RTVE. Es la última ley en vigor, de carácter común.

  Esta ley establece "los principios mínimos que deben inspirar la presencia en el sector audiovisual de organismos públicos prestadores del servicio público de radio, televisión y servicios interactivos", y establece, por tanto, las condiciones de los contenidos emitidos por RTVE o disponibles en sus plataformas audiovisuales.

  Garantiza los derechos de los ciudadanos en cuanto a contenidos plurales e inclusivos, los derechos de los administradores de servicios de comunicación audiovisual, publicidad, libertad de empresa y régimen jurídico. Se valoran también nuevas formas de comunicación audiovisual y la libre competencia y pluralismo entre medios televisivos y radiofónicos.

  Esta ley se alinea con la legislación europea existente (Comunicaciones, Directivas, Decisiones y Recomendaciones) sobre servicios públicos de radiodifusión. Se trata también la Autoridad Audiovisual estatal y su regularización junto con el Consejo Estatal de Medios Audiovisuales (CEMA)

  Establece los requerimientos en los ámbitos que se resumen brevemente a continuación, conforme a la Comisión Nacional de Mercados y Competencia [26]:

Tabla 2.3: Resumen de las exigencias básicas establecidas por la Comisión Nacional de los Mercados y Competencia para control de contenidos en medios audiovisuales.

Ámbitos	**Requisitos básicos**
Menores	Derecho a que no se emplee su voz e imagen sin su consentimiento o del representante legal. Prohibida difusión de sus datos. Horario de protección general (no contenidos para mayores de 18 años) entre las 6h y 22h. Franjas reforzadas (no contenidos para mayores de 12 años): 8 a 9h, y 17 a 20h, fin de semana 9 a 12h. Programas esoterismo y paraciencias 22h-7h, juegos de azar y apuestas 1h-5h. Igualmente no se puede emitir publicidad de productos de autoimagen y culto al cuerpo (tratamientos estéticos o quirúrgicos, productos adelgazantes, etc.)
Publicidad	Límites de tiempo para autopromoción (5 minutos/h), mensajes publicitarios y televenta (12 minutos/h), número de interrupciones publicitarias, principio de identificación y separación de programación y publicidad, carácter no publicitario de anuncios de servicio público o carácter benéfico.
Accesibilidad	Accesibilidad universal para personas con discapacidad visual o auditiva, conforme a las posibilidades tecnológicas. Diversidad de medios (interpretación en lengua de signos, audiodescripción de los contenidos, subtitulación). Estos medios deben cumplir un porcentaje semanal: - Canales en abierto: subtitulado 75%, audiodescripción 2h/semana, lengua de signos 2h/semana - Canales servicio público: subtitulado 90%, audiodescripción 10h/semana, lengua de signos 10h/semana

2.3.2. Datos de carácter personal

Dentro de la legislación aplicable a la protección de datos de carácter personal:

- **Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal (LOPD)** [27]. Cumplía un propósito de garantizar y proteger las libertades públicas y derechos fundamentales de las personas físicas en cuanto al tratamiento de los datos personales. Ha estado en vigor hasta hace un año, siendo derogada en diciembre de 2018 con la siguiente ley de la que se hace mención:

- **Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales (LOPD-GDD)** [28], adaptación del Derecho interno español al Reglamento General de Protección de Datos (RGPD), normativa de ámbito europeo. Esta reforma de la protección de datos ha entrado en vigor en 2018, aplicable sobre todos los organismos, empresas, instituciones y organizaciones que manejen cualquier tipo de información personal, se ha percibido fuertemente por los usuarios, que han debido actualizar su concesión para la gestión de sus datos y consentimiento para el almacenamiento. (Tabla 2.4)

Tabla 2.4: Principales ámbitos de la LOPD de la Unión Europea.

Ámbitos	Requisitos básicos
Coordinación	Se aplica un único conjunto de leyes para todos los estados miembros. Cada estado miembro establece una autoridad para supervisión (SA), capaz de escuchar e investigar denuncias de forma independiente, sancionar las infracciones administrativas que puedan identificarse, etc. Las SA de cada estado trabajarán de forma sincronizada, coordinando operaciones conjuntas y proporcionando asistencia mutua. Las empresas multinacionales tendrán una autoridad principal encargada de supervisar, a modo de "ventanilla única".
Responsabilidad	Requisitos de notificación incorporan duración del almacenamiento de los datos personales e información de contacto del controlador de los datos. Automatización de toma de decisiones es discutible (p.ej. creación de perfiles automáticos), deben garantizarse medidas de protección de datos por diseño y protección de datos por defecto.
Base legal para tratamiento	Sólo podrán tratarse los datos si existe una o más bases legales que lo justifiquen: - Consentimiento explícito del interesado - Tratamiento necesario para ejecución de un contrato - Tratamiento necesario por obligación legal sobre controlador - Tratamiento necesario para proteger intereses vitales - Tratamiento necesario para realización tarea interés público o ejercicio de autoridad conferida al controlador. - Tratamiento necesario por fines de intereses legítimos, salvo que dichos intereses sean contrarios a derechos y libertades fundamentales que requieren protección de datos personales, en particular siendo el interesado un niño.
Consentimiento	Siendo el consentimiento la base legal, debe ser explícito en los datos recopilados, así como los fines para los que se utilizarán estos datos. El consentimiento para niños debe ser concedido por un padre o tutor, y verificable. El consentimiento debe ser demostrable por el controlador, y puede ser retirado.
Seudonimización	El RGPD de la Unión Europea recomienda cifrado o encriptación para máxima seguridad de los datos almacenados, y para evitar consecuencias en el caso de fugas de información. La seudonimización consiste en el reemplazo de ciertos datos sensibles por códigos artificiales únicos para cada campo o grupo de campos reemplazados. Es alternativa a la anonimización de datos, en que se pierde toda traza de información personal del individuo del que procedían.

2.3.3. Referencias

Para la elaboración de este trabajo se han consultado numerosas fuentes bibliográficas y documentos accesibles en internet. Por ello, se considera aplicable la legislación vigente de propiedad intelectual.

- **Ley 2/2019, de 1 de marzo** [29], que modifica la anterior Ley de Propiedad Intelectual (aprobada en el Real Decreto Legislativo 1/1996, de 12 de abril) e incorpora al ordenamiento jurídico español la Directiva Europea 2014/26/UE del Parlamento Europeo y del Consejo, de 26 de febrero de 2014.

2.3.4. Licencias software.

Las condiciones de utilización de los sistemas software son distintas en función de las condiciones del contrato que se establece entre usuario y proveedor a través de los términos legales y condiciones en el momento de la adquisición o descarga.

Tabla 2.5: Tipos de licencia software.

Licencia	**Condiciones**	
Software libre	**Disponibilidad para utilización, modificación, copia y distribución.**	
Copyleft	Disponibilidad para utilización libre, pero normas establecidas para la modificación, copia y distribución	
Licencias abiertas particulares	Licencia Pública General GNU (GNU General Public License GPL) para paquetes GNU dentro de Linux	Linux
	Debian Free Software Guidelines (DFSG), para Debian	Debian
	Open Source Iniciative de Debian	Debian
Software con dominio público	Software sin copyright, sujeto a posibles restricciones adicionales impuestas por el autor para redistribución o modificación	
Software semi-libre	No es libre para utilización, pero se permite la copia, distribución y en ocasiones modificación	
Freeware	**Disponibilidad para redistribución, pero no modificación. Programas gratuitos pero no modificables. Código fuente no disponible**	
Shareware	Disponibilidad para redistribución, pero uso a través de adquisición. Recursos restringidos o limitados en el tiempo hasta que se efectúa el pago. Abarca licencias adware (anuncios sobre programa principal), trial (prueba temporal) y demo (sólo algunas funciones disponibles).	
Software propietario	**Prohibida su copia, redistribución o modificación, sin autorización del propietario o pago de la nueva. Código fuente no disponible**	
Software comercial	**Desarrollado para lucrar su utilización. Puede ser abierto (código fuente disponible) o propietario (no disponible)**	

En este proyecto se han empleado programas de software libre y freeware, que no han sido modificados en su código fuente. También el paquete gratuito de dos software comerciales, con duración temporal y funciones limitadas, que tampoco han sido modificados en su código fuente ni redistribuidos, por lo que se cumplen sus condiciones de utilización. Las librerías empleadas en el desarrollo del código se adscriben al mismo tipo de licencia que los soportes en que se han desarrollado.

Los programas empleados se enumeran en el apartado 3.2 “Herramientas utilizadas”, y su clasificación en licencias en el apartado 5.4 “Presupuesto”.

2.4. Marco socioeconómico

Para una mejor comprensión del marco socioeconómico, a continuación, se presenta un análisis de Debilidades, Amenazas, Fortalezas y Oportunidades (DAFO) del entorno[8]:

2.4.1. Fortalezas

En el paradigma actual, las **tecnologías de procesamiento de lenguaje natural** (*NLP*) están en pleno auge, y las técnicas y modelos que se estudian están en constante desarrollo. Desde el punto de vista de las **empresas**, son tecnologías relevantes porque:

- Permiten extraer una información más completa a partir de textos u oraciones de los usuarios. Permiten unos procesamientos posteriores más cercanos a la respuesta que daría un ser humano. Por ejemplo, asistentes virtuales de atención al usuario, soporte de incidencias, gestión de situaciones básicas.

- Permiten automatizar tareas, por ejemplo, la comprensión, resumen y clasificación de mensajes recibidos en un buzón de reclamaciones, dudas y sugerencias. La identificación del tipo de mensaje del que se trata, del tema particular al que hace referencia, identificación de la intención u opinión a ese respecto, extracción de ideas principales en el mensaje, identificación de datos particulares del usuario.

- Capacidad de procesamiento: potencia para manejar vastas cantidades de información, velocidad, fiabilidad (no hay error humano), consistencia en los resultados.

- Permiten extraer relaciones entre entidades, conceptos o términos que aparecen en un texto, que no serían intuibles por un ser humano o perceptibles a simple vista. En concreto, estos modelos de interrelación pueden desarrollarse con técnicas de *Deep Learning* y *Big Data*, y permiten desarrollar estrategias no evidentes para una persona.

  Por ejemplo, en esta entrevista [30] a Jared Kushner, responsable de la campaña de redes sociales de Donald Trump, explica cómo mediante el seguimiento de los intereses de los potenciales votantes por ciertas series y películas, encontraron relaciones entre estos intereses y las preocupaciones políticas de los ciudadanos. Empleando estos datos, se desarrolló la campaña dando énfasis a determinados temas en ciertas zonas geográficas.

- En las compañías internacionales, posibilidad de traducciones más fieles al sentido original de las frases, al captar más profundidad en el mensaje. Traducciones automáticas, posibilidad de soporte web en un abanico de idiomas más amplio que mediante empleados traductores (normalmente manejan 2 o 3

[8] *El presupuesto para la solución propuesta se detalla en el apartado "5. Organización", así como los detalles técnicos de la misma se desarrollan en el apartado "3. Diseño solución técnica".*

idiomas, frente a todos los que puede gestionar un software bien desarrollado), correctores de estilo o gramaticales…

- Facilidad para la compatibilidad entre distintos sistemas o lenguajes. Al extraer características intrínsecas y no anclarse en la enunciación concreta, puede permitir una mayor transversalidad entre tecnologías y plataformas. Conversores.

- Identificación de datos personales del usuario, tracking de gustos, intereses, patrones de comportamiento. Cruce de información entre distintas plataformas, comportamientos sociales. Nuevos ámbitos de interés de los usuarios en los que hacerse presentes para captar clientes potenciales. Estudios de mercado, estrategias comerciales. Identificación de oportunidades potenciales o de movimientos en la competencia.

Por todos estos motivos, y por el propio auge de estas tecnologías que ha habido en consecuencia, numerosas **empresas** tienen en sus equipos departamentos de desarrollo en tecnologías *NLP* y *Machine Learning*, o bien financiación de proyectos externos en universidades u otras compañías. Estas investigaciones se están desarrollando en numerosas universidades de todo el mundo (sin ir más lejos, se pueden consultar en la Tabla 2.2 diversas tecnologías *NLP* desarrolladas desde universidades), en compañías especializadas en este ámbito (tanto a pequeña escala, con plantillas de menos de una veintena de trabajadores, como a grandes empresas), y también en potentes compañías tecnológicas (departamento DeepMind de Google, OpenAI de Elon Musk (fundador de Tesla), departamento FAIR de Facebook, o los de IBM, Microsoft), y, por ejemplo, compañías dedicadas al desarrollo de coches autónomos con tecnologías de reconocimiento de voz y *NLP* para comprensión de instrucciones.

Dentro del caso concreto de las **plataformas de contenidos multimedia**, como por ejemplo RTVE, el interés de estas tecnologías es más para tareas de documentación de archivo e indexación de contenidos:

- El primer objetivo de la implementación de estas tecnologías para los datos de la API de RTVE es la identificación de las temáticas de que trata cada vídeo o parte de un vídeo (por ejemplo, en un archivo de telediario, separar en los temas de los que se habla: política, sociedad, ecología, educación, deportes…). Es útil para la catalogación de los archivos, y para indexar los contenidos y que sean más fácilmente localizables.

- De la misma manera, se desarrolla identificación de las entidades y personajes concretos que se mencionan en él (personas, lugares, organizaciones). Esto es útil, por ejemplo, en el caso de que un personaje importante fallezca, o haya una noticia importante sobre cierta organización o lugar, de forma que puedan recuperarse del histórico todas las referencias a este personaje u organización, y pueda elaborarse un monográfico o documental sobre este tema.

- En un paso posterior, pueden combinarse estos sistemas con tecnologías de seguimiento de la acción del usuario (monitorizar en qué vídeos de los sugeridos entra el usuario, en cuáles comenta, qué temáticas o partes de vídeo le interesan,

con qué otros vídeos o noticias se relacionan). Personalización de los contenidos, oferta única y adaptada a los gustos e intereses del consumidor.

- En este momento, estas tareas las realizan personas especializadas en documentación: archiveros, bibliotecarios, documentalistas, o bien periodistas o editores. La automatización de estas tareas aligeraría la carga de trabajo, y permitiría sistemas más rápidos y potentes.

Los dos primeros puntos de esta lista también tienen interés para archivos de librerías, lugares con mucha documentación histórica, o para la **investigación**, por ejemplo, al crear o buscar artículos científicos. Un etiquetado automático de los archivos y una extracción automática de las palabras clave (*keywords*) podría hacer que su índice de impacto creciera, por la prioridad que tienen ciertos términos o *keywords* frente a otros que a una persona podrían parecerle igual de significativos o incluso mejores. También tiene sentido que, dado que los buscadores de internet funcionan computacionalmente, la asignación de etiquetas y *keywords* sea consistente con las reglas de búsqueda.

Desde el punto de vista de los **usuarios**, estas tecnologías son deseables por:

- La novedad que suponen, la diferenciación frente a otros productos de la competencia (smartphones, por ejemplo). Por el hecho de ser tecnologías pioneras y no implantadas en todos los dispositivos, ni al alcance de todos.

- Por la comodidad de uso, las facilidades añadidas que se derivan de una configuración más personalizada de los servicios y funcionalidades de los aparatos. Nuevas opciones, y formas más cómodas de gestionar los archivos, aplicaciones, herramientas…

- En el caso de aplicarse unas correctas políticas de protección de datos y privacidad de los usuarios, pueden proporcionar un almacenamiento seguro de muchos datos que tal vez en otras circunstancias no se confiarían a dispositivos digitales, incluyendo datos personales de los que el usuario no es consciente, sino que han sido obtenidos de forma automática por los sistemas.

En un ámbito común entre los usuarios y empresas o universidades, estas tecnologías son relevantes para la **investigación** porque:

- Permiten realizar análisis más profundos de los datos de que se dispone, permiten manejar vastas cantidades de datos y procesarlos con una rapidez muy superior a la que se podría obtener por métodos más convencionales.

- Permiten encontrar artículos, documentos, artículos científicos, *abstracts*, libros, y muchas referencias de una forma más centrada en la idea o ámbito del que se desea la información, que en coincidencias léxicas entre los términos de búsqueda. Búsquedas semánticas, relaciones con documentos que no comparten palabras, pero sí significado. Búsquedas semánticas.

- Permite contrastar información de diversas fuentes, incluso cotejar versiones de una noticia o acontecimiento, para contrarrestar el sesgo que pueda mostrar el autor o empresa que lo ha publicado en favor de una versión.

2.4.2. Debilidades

- Con todas las nuevas tecnologías, un punto crucial es la gestión de los datos de usuario, y las políticas de protección de datos y privacidad. En este sentido, existe una **vulnerabilidad** a ataques para la obtención de datos sensibles de los usuarios: ventas o cesiones a terceros que terminen en lugares insospechados, robos de información, ataques cibernéticos, y las consiguientes denuncias potenciales de los usuarios por la mala gestión de los datos. La centralización de los datos de los usuarios y su recopilación se convierten en un fuerte atractivo para ataques y venta de información (ataques para fugas de información, también espionaje industrial).

- Visión parcial de los resultados: las conclusiones que se extraigan de los análisis nunca serán 100% fiables, y, sin embargo, deberá ser la información sobre la que se fundamenten los siguientes procedimientos y las decisiones posteriores. Resultados sujetos a **error**.

- Relacionado con esto, en los sistemas de *Topic Modeling* entra en juego el criterio del programador para validar el acierto o desatino en los *topics* que la máquina ha obtenido. Por ser una **tecnología no supervisada**, las **evaluaciones** son **subjetivas**, ya que no existen métricas de medición de lo adecuado o irrelevante de los términos dentro de un *topic*, por ejemplo. En la extracción de entidades, los resultados son más fácilmente evaluables, puesto que en el texto hay un número concreto de entidades, que el sistema localiza o no, y un número de "no-entidades", que el sistema puede descartar acertadamente o confundir con entidades.

- También hay una posibilidad de **sobreajuste** sobre el caso particular que se emplee de ejemplo. El sobreajuste (en inglés, *overfitting*) es un problema del aprendizaje automático que se produce cuando el sistema, tratando de optimizar la solución para no fallar nunca, aprende demasiado sobre la particularidad del problema, por lo que pierde capacidad de generalización. Siendo así, los resultados que se obtendrían evaluando sobre las muestras de entrenamiento serían muy buenos, pero al aparecer un caso nuevo sin las particularidades del anterior, el sistema fallaría sistemáticamente.

  Esto podría evaluarse empleando técnicas de **validación cruzada**, en que se dividen las muestras disponibles en "n" partes (p.ej., 5), y se emplean todas menos una para entrenar el modelo. Se evalúa el modelo con la parte restante, se reajusta, y entonces se reentrena con otra combinación de 4+1 partes (p.ej., se toman las dos primeras partes y las dos últimas y se evalúa con la de en medio). Así, se entrena y valida con distintas combinaciones de los datos, y se contrarresta el sesgo que pueda haber.

- En los procedimientos desarrollados con sistemas de Machine Learning, dificultad de interpretación de los resultados, **opacidad** del sistema. El programador no conoce cuáles son las características que han llevado al modelo a extraer cierta conclusión.

- Desde el punto de vista del usuario, pérdida o cesión del **control de sus datos**. Se proporciona mucha información al sistema, lo cual según el análisis de las fuerzas de Porter [31] da poder negociador a las tecnologías.

- También es una debilidad la **falta de experiencia** en este ámbito. Siendo sistemas desarrollados muy recientemente, la experiencia en su uso y gestión es reducida, por lo que la **inversión** que supone en el desarrollo de estos sistemas puede ser una apuesta arriesgada, que puede tener muy baja rentabilidad. Como parte positiva de esto, en todo caso la inversión que se haga para el soporte de estos sistemas sería reutilizable para otras tecnologías, ya que los equipos (ordenadores, principalmente) suelen ser de propósito general.

2.4.3. Amenazas

Dentro de este sector, las principales amenazas son los productos sustitutivos de empresas de la competencia. El desarrollo de mejores tecnologías, o con más rapidez o mejor rentabilidad para el usuario puede marcar la diferencia entre el éxito de una tecnología o su completo fracaso.

2.4.4. Oportunidades

En el momento actual, ninguna de las herramientas *NER* que existen en el mercado ha alcanzado el máximo desarrollo al que podía dar de sí, sino que todas están en crecimiento. Esto constituye una oportunidad porque el resultado aún no está decidido, y aún empresas o grupos de investigación recién llegados pueden desarrollar sistemas punteros en este campo. También, al haber tantas opciones abiertas, la investigación tiene mucho que desarrollar en este ámbito, y hay por lo tanto mucha libertad para innovar los sistemas.

Casi todos los sistemas o herramientas, ya sean analógicos o digitales, siguen una curva de crecimiento en su producción, explotación y uso. Con las mejoras en ellos, se abaratan los costes y a su vez mejoran las prestaciones, por lo que el consumo crece. Sin embargo, esa curva de crecimiento tiene un punto en el que alcanza su máximo crecimiento, lo máximo que podía dar de sí esa tecnología. Suele coincidir con la aparición de sistemas sustitutivos superiores a los que se tenían, de forma que los antiguos no son capaces de competir en prestaciones ni siquiera con el bajo precio que van tomando.

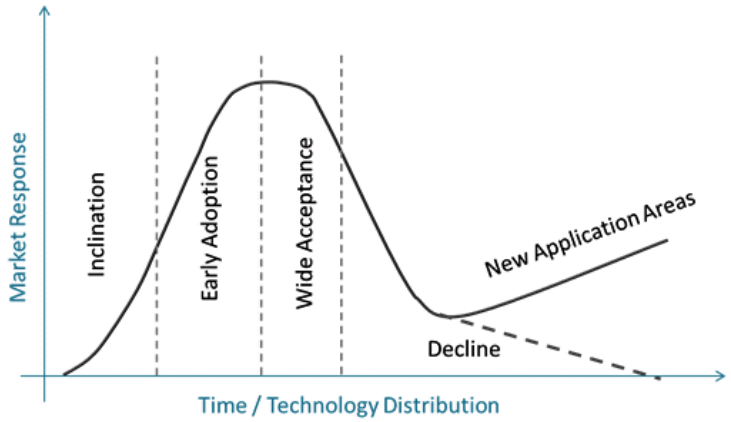


Fig. 2.8: Ciclo de vida de una tecnología. Fuente: Greyb [32]

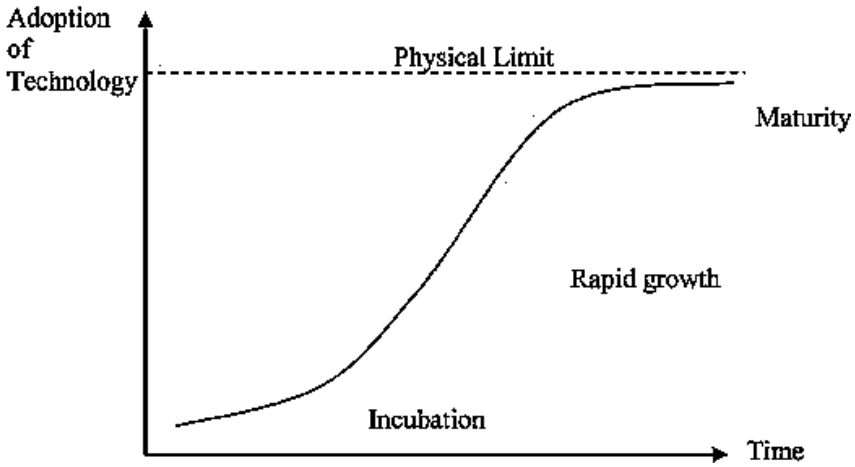


Fig. 2.9: Curva en S del crecimiento y difusión de una tecnología. Fuente: ResearchGate [33]

Esta situación, en el ámbito de NLP, y también dentro del *Machine Learning* y gestión de *Big Data*, aún está lejos de suceder, puesto que estamos en el despegue de estas tecnologías. Estos sistemas se encontrarían en la zona de máximo crecimiento de ambas curvas.

A corto y medio plazo las principales oportunidades son en el desarrollo y utilización de sistemas de *Deep Learning* (aprendizaje profundo), que permiten nuevas aproximaciones y soluciones innovadoras tanto a problemas ya resueltos como a nuevos retos antes irresolubles. Típicamente se implementan a través de Redes Neuronales Multicapa, dentro de la inteligencia artificial (*AI, Artificial Intelligence*), también se pueden desarrollar soluciones que optimizan la resolución de problemas. Por ejemplo: dentro de las herramientas *LDA* que realizan *Topic Modeling* a través de modelos estadísticos, se están desarrollando sistemas implementados a través de redes neuronales (*Autoencoding Variational Inference For Topic Models, AVI.TM*, referencia en el apartado 2.2.2) para la solución de este problema.

3. DISEÑO SOLUCIÓN TÉCNICA

3.1. Introducción y diagrama de bloques del sistema

El proyecto consiste en un sistema de correlación de los resultados de la extracción de entidades nombradas (*NER*) con los términos obtenidos del modelado de *topics* (*TM*).

Para ello, se han desarrollado independientemente los dos módulos ya mencionados, relacionándose al final a través de los resultados de uno y otro.

La primera parte del trabajo se desarrolló dentro del departamento de Experiencia del usuario de RTVE, con la tutorización complementaria de D. Manuel Gómez Zotano, director del departamento. La continuación de esos análisis y procedimientos se desarrolló de forma independiente por la estudiante, y con la tutorización de Simón Roca Sotelo, profesor del departamento de Teoría de la Señal de la Universidad Carlos III de Madrid.

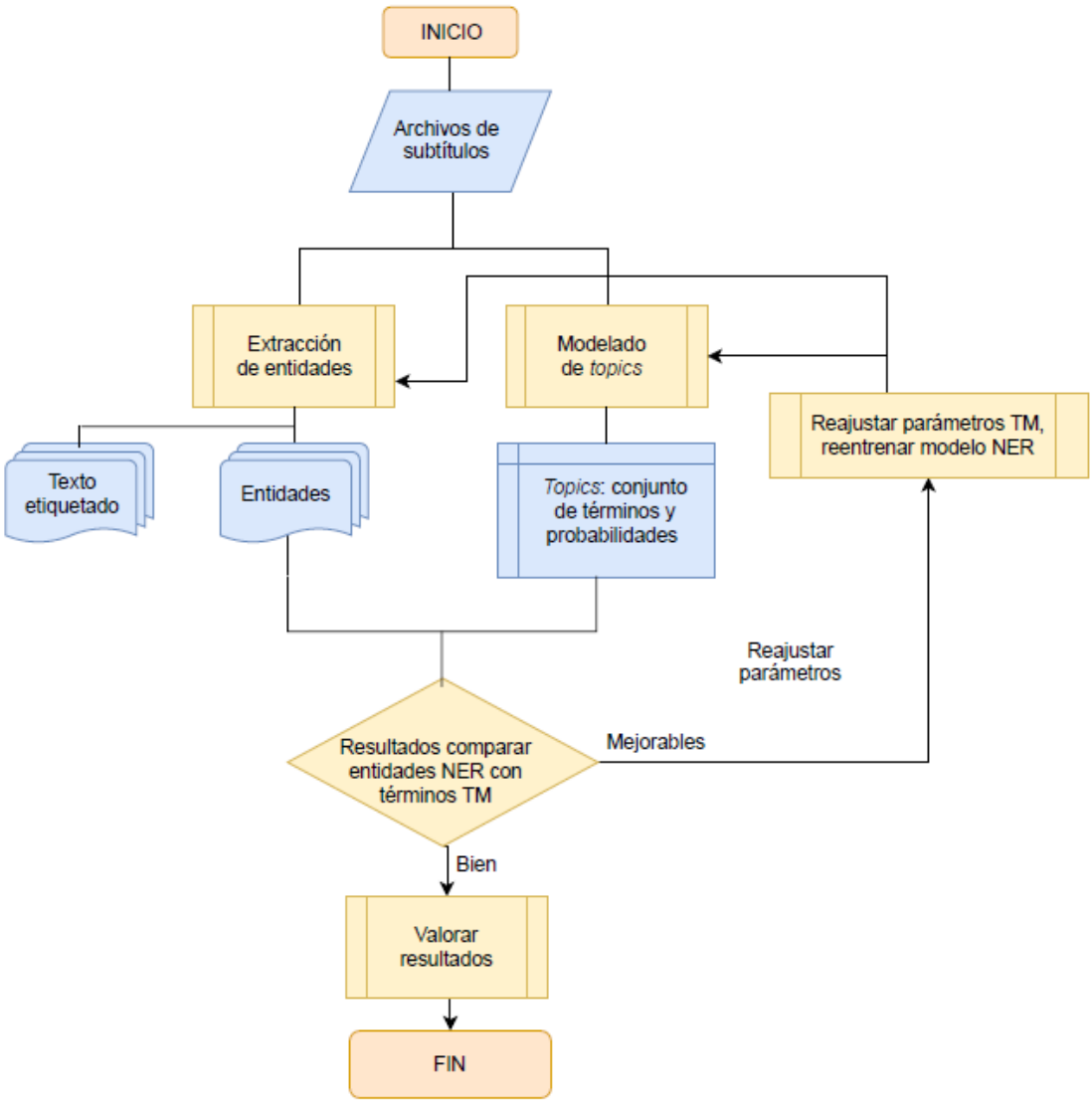


Fig. 3.1: Esquema de funcionamiento general.

3.2. Herramientas utilizadas

Para el desarrollo de este sistema se han empleado distintas técnicas y herramientas en varios lenguajes:

Tabla 3.1: Tabla de módulos y librerías empleados en cada lenguaje y sección durante este trabajo, y breve descripción de los mismos.

Bloque		**Sistemas**	**Descripción**
Web	**Cliente HTTP**	Node JS	Entorno de ejecución JavaScript para la capa de servidor, basado en el lenguaje ECMAScript, con arquitectura orientada a eventos
NER	**Java**	Lenguaje de programación concurrente, orientado a objetos, diseñado para tener el mínimo número posible de concurrencias.	
		Apache OpenNLP	Conjunto de herramientas basadas en aprendizaje automático para el procesamiento de texto en lenguaje natural.
	R	Entorno y lenguaje enfocado al análisis estadístico, muy empleado en investigación científica, minería de datos, investigación biomédica, bioinformática y matemáticas financieras.	
		NLP	Paquete de CRAN para procesamiento de lenguaje natural.
		openNLP	Interfaz R para las herramientas de Machine Learning NLP desarrolladas en Apache OpenNLP
		RJava	Interfaz de bajo nivel para la máquina virtual de Java, permite la creación de objetos, métodos de llamada y campos de acceso
		RWeka	Algoritmos de Machine Learning y tareas Data Mining en Java.
		Magrittr	Paquete R para disminuir el tiempo de desarrollo y para mejorar la legibilidad y la capacidad de mantenimiento del código, permite encadenar resultados en estructura de tuberías (*pipelines*).
		Qdap	Automatización de tareas asociadas con análisis cuantitativo de discurso (recuento de frases, palabras, turnos de palabra, sílabas…).
		openNLP models	Modelos pre-entrenados de entidades para openNLP. Existen modelos de fecha, persona, organización, lugar, tiempo, etc. en varios idiomas
	LUIS (Microsoft Azure)	Servicio basado en Machine Learning para desarrollar comprensión lingüística natural para aplicaciones, bots y dispositivos IoT.	
	NLU (IBM Watson)	Funciones avanzadas de análisis de texto para extraer entidades, relaciones, palabras clave, roles semánticos y más.	

Bloque		Sistemas	Descripción
TM	**Python**	Lenguaje de programación interpretado que busca la legibilidad y fácil interpretación humana. Soporta orientación a objetos, programación imperativa, y en menor medida, programación funcional.	
		Spacy	Librería de código abierto para NLP en Python
		NLTK	Conjunto de librerías y lenguajes para NLP simbólico y estadístico
		Gensim	Librería de código abierto para TM no supervisado y NLP mediante Machine Learning (aprendizaje automático)
		PyLDAvis	Librería para la visualización de resultados

A continuación se detalla el proceso seguido para cada parte del proyecto.

3.3. Datos empleados.

Este trabajo ha empleado como ficheros un set de archivos de subtitulado de telediario y programas de RTVE. Se han procesado los archivos de 10 temáticas, a través de 1.297 archivos con 100 frases cada uno. Para el sistema *NER* la identificación de entidades puede realizarse desde un único archivo, y de forma independiente en todos los que se desee, mientras que para *TM* se requiere un elevado número de documentos, aunque sean de menor longitud.

Para interactuar con la API de RTVE[9] se emplean peticiones HTTP, y se manejan documentos JSON. En este bloque, se guardan los ficheros codificados con los subtítulos, y se eliminan los códigos y marcas de tiempo. La codificación de esta parte del proceso se realizó en **Node JS**, utilizando el entorno gráfico de Visual Studio Code.

3.4. *NER*

Antes de comenzar a implementar el sistema, se realizó un estudio de las principales tecnologías dentro de este campo (detalles en la Tabla 2.2). Además del estudio comparativo a través de documentación, artículos y páginas oficiales, se seleccionaron 4 tecnologías con las que realizar un reconocimiento de entidades, para así poder seleccionar la que proporcionase unos mejores resultados.

Los sistemas que se proponía evaluar fueron los sistemas *NER* en los lenguajes Java y R, y los módulos de procesamiento de lenguaje natural de Microsoft y de IBM.

En **Java** y en **R** se ha empleado un procedimiento de aprendizaje supervisado, consistente en los siguientes pasos:

[9] API de RTVE accesible desde http://www.rtve.es/api/tematicas

1. Preprocesado de los archivos de texto para eliminar los códigos que pudieran quedar, y para prevenir errores en el procesamiento posterior.

2. Anotación de palabras y frases en el texto. Para que funcione correctamente el *NER*, el sistema debe saber dónde termina una palabra y comienza la siguiente, y lo mismo con las oraciones. En las librerías de *NLP* de Java y R vienen incluidos los modelos de anotación de palabras y frases.

3. Procesamiento de los archivos empleando el modelo NER pre-entrenado para personas, lugares y organizaciones, que pueden descargarse con unas librerías. Estos modelos son comunes en *NER*, y por eso existen versiones ya entrenadas. Identificación de las entidades de persona, lugar y organización.

4. Extracción de esas entidades: por cada documento que ha sido procesado, se crea un archivo con la lista de entidades localizadas, y a su vez se crea otro archivo con el texto esta vez marcado con las etiquetas.

5. Llegados a este punto, se pueden evaluar los resultados: se comparan las entidades halladas por el sistema con las que realmente existen, para así poder obtener unas métricas del error desde cada uno de los sistemas.

6. Para poder comparar los resultados obtenidos con las soluciones reales, manualmente se etiquetan todas las entidades de interés en el texto. En otras situaciones podrían existir archivos previamente etiquetados con las entidades, pero al estar trabajando con los archivos de RTVE no están disponibles estas plantillas con la solución. También podría buscarse algún otro sistema que realice este proceso con más agilidad y buenos resultados, pero para asegurar que las soluciones tengan una fiabilidad lo más fiable posible, se procede a hacerlo manualmente.

7. Obtenidos ambos archivos etiquetados (el resultado del procesamiento automático y el etiquetado manualmente), se comparan el número y características de las entidades correctamente etiquetadas y de las que no. También, mediante observación de las entidades etiquetadas incorrectamente o no etiquetadas, pueden percibirse patrones o situaciones que hayan conducido a la máquina a actuar de esa manera. Por ejemplo, palabras con mayúscula en medio de una oración, o conjuntos de palabras etiquetados como una sola entidad cuando son dos.

   Se pueden obtener métricas del acierto o error en *NER* (ver Tablas y resultados en apartado 4).

Pese a haber analizado las entidades que más error producían en ambos sistemas, y haber comparado los resultados en ambos, no hemos llegado a ninguna conclusión que justifique por qué los resultados para el sistema implementado en R y en Java son tan dispares, siendo los mismos archivos, y el mismo proveedor de los modelos (openNLP).

Estos dos sistemas permiten el re-entrenamiento de los modelos, por lo que se pueden repetir los pasos anteriores empleando esta vez un modelo nuevo, basado o bien en las etiquetas que asignó el sistema en la primera iteración, o bien en las etiquetas asignadas manualmente.

Así puede iterarse las veces que se considere conveniente, y puede observarse la evolución del error en función del número de pasadas. Aunque pudiese parecer que mejores serán los resultados conforme más veces se repita el entrenamiento, esto es contraproducente. (Consultar el glosario para la explicación de sobreajuste y subajuste en el entrenamiento).

En la evaluación de los módulos *NLP* de Microsoft e IBM no se ha llegado a alcanzar resultados, debido a lo particular de los procedimientos propietarios. Sin embargo, se ha llegado a comprender el procedimiento general, que se detalla a continuación:

Microsoft Azure "Language Understanding Intelligent Service" (LUIS):

1. Para trabajar dentro del sistema LUIS se requiere una cuenta de Microsoft, Outlook o una cuenta corporativa de la empresa. Se necesita un plan de contratación de los servicios disponibles, en función de la carga computacional deseada, duración del proyecto y algunos otros factores. Por ser una evaluación de los sistemas y no preverse una larga duración, se selecciona el plan gratuito, que tiene unas prestaciones limitadas en tiempo y potencia.

2. Una vez creada la cuenta, se puede comenzar a desarrollar. El sistema LUIS se basa en "*intents*", "*utterances*" y "*entidades*" (intenciones, declaraciones y entidades). Los *intents* son intentos de realizar una acción, los *utterances* son entradas de texto que LUIS puede interpretar, y las *entidades*, los sujetos (personas, lugares, organizaciones, etc.) relevantes para la cuestión. El sistema muestra un pequeño tutorial del funcionamiento de estos tres conceptos.

3. El primer paso consiste en crear una aplicación LUIS, que permite asignarle un nombre, una "cultura" (idioma del texto) y una descripción.

4. Dentro del escritorio de la página se pueden crear los *intents* y entidades que se deseen para el proyecto, y revisar las *utterances*.

5. A continuación, se debe añadir el módulo de NER para incluirse en la aplicación que se está creando, y a partir de aquí se puede comenzar a desarrollar código para obtener los resultados.

IBM Watson "Natural Language Understanding" (NLU):

1. El procedimiento dentro del sistema de Watson es muy parecido: en primer lugar, se crea una cuenta para acceder al gestor de aplicaciones de IBM.

2. Dentro del panel de control de los proyectos se debe seleccionar el módulo o proyecto deseado, en este caso, "Watson Natural Language Understanding Basic". Por lo antes explicado, se selecciona también aquí el plan gratuito, aunque tenga funcionalidades limitadas.

3. A continuación, aparecen los ajustes para la creación del nuevo proyecto: nombre, ruta host, dominio (*clusters* servidores disponibles en distintas regiones del globo,

por ejemplo, Europa, Asia del Este, Asia del Oeste, etc.), lenguaje en que se va a desarrollar (Node JS o Python+Flask).

4. Una vez creado el nuevo proyecto, el entorno permite descargar un código de prueba para la aplicación, para ejecutarlo localmente (a través de Docker) y desarrollar localmente sobre IBM Cloud.

5. Otra forma de seleccionar los módulos consiste en la navegación por las categorías (infraestructura, plataforma, etc.), y, dentro de plataforma, seleccionar los bloques deseados (API, internet de las cosas, datos y análisis, inteligencia artificial Watson, etc.). Se deben seleccionar los servicios que se desee incluir en el proyecto.

Como ya se ha comentado, por lo particular de cada una de estas dos tecnologías, no se ha conseguido demasiado avance en esta línea ni se han podido extraer conclusiones sobre las prestaciones y características de estos sistemas en cuanto la implementación de NER.

3.5. *TM*

Dentro de la parte de modelado de *topics* se ha realizado en primer lugar una tarea de investigación y documentación del paradigma actual (ver apartado 2.2.3 para más detalles).

Posteriormente, se ha procedido al desarrollo de un sistema *TM* en **Python**, que, a partir de 1.297 documentos de unas 100 frases de longitud, obtiene los términos más recurrentes en documentos que tratan de una temática o que componen un *topic*.

El desarrollo ha seguido las siguientes etapas:

1. Pre-procesamiento de los documentos con que se iba a trabajar. Se disponía de un número inferior de documentos, de longitudes muy dispares, y separados por categorías temáticas según los códigos de la API de RTVE (telediario, documentales, Águila Roja, Cuéntame, etc.). En lugar de esto, se ha optado por confeccionar documentos únicos dentro de cada temática, con todos los archivos disponibles de esa temática. Con ellos preparados, se ha procedido a dividir esos grandes documentos en documentos de una longitud de 100 frases (suponen entre 3 kB y 21 kB, en función de la longitud de las frases, en relación también con la temática de la que se trate).

2. Limpiar el texto y tokenizar (dividir en unidades o *tokens*, que serán las palabras). Se puede *tokenizar* a través del tokenizador de Regexp (expresiones regulares, patrones en las palabras), o a través de Spacy.

3. Para identificar los significados de las palabras, sinónimos, antónimos, etc., en inglés se puede importar la librería Wordnet, que es una página referente de tipo diccionario del inglés. Esta librería permite también "lematizar" (tabla 2.1 de subtareas de NLP), que consiste en reducir las palabras o *tokens* a la forma más básica de palabra que tenga significado y aparezca en el diccionario. (No debe confundirse con la poda o *stemming*, que reduce la palabra a su lexema o raíz,

aunque no tenga significado como palabra en sí misma). Sin embargo, por ser con textos en español, la lematización se hace desde la librería Spacy previamente importada, que separa en palabras y obtiene el "lema" y la función gramatical de cada una.

4. Se filtran también los términos que no son significativos (*stopwords*: adverbios, preposiciones, pronombres, artículos), que no aportan información semántica y sólo aparecen para estructurar la frase, pero que no tienen significado a la hora de caracterizar los *topics* dentro de un texto. Se eliminan asimismo las palabras de menos de 4 letras, que principalmente son artículos, pronombres y preposiciones. Se filtran también las palabras que terminan en "-ar", "-er", "-ir", que son verbos y no siempre aportan mucha información.

5. Ya está el texto preparado para hacer el TM. Dentro del directorio, se leen todos los archivos de la carpeta, y se elabora una lista de *tokens* o palabras presentes en cada archivo.

6. Empleando el modelo *LDA* del módulo "corpora" (plural de "corpus"), se crea un diccionario a partir de la lista de palabras del archivo. Este diccionario sirve para crear el "corpus", el conjunto de datos con el que se va a trabajar para crear el modelo de *topics*.

7. El módulo *gensim* permite seleccionar el número de pasadas al corpus para entrenar el modelo *LDA*, y el número de *topics* que se desean visualizar. Ahora ya se pueden obtener los porcentajes de aparición de cada *topic* dentro de un documento, y ver cuál es el *topic* principal y su probabilidad. Se relaciona también esta información con los términos (en inglés, "*terms*" que caracterizan cada *topic*).

8. Por último, desde *pyLDAvis* podemos visualizar la lista de palabras que caracterizan los *topics*, pero también podemos ver un diagrama (Fig. 3.3) de la interrelación de los *topics* entre sí en el conjunto de documentos (a la izquierda), y de los términos dentro de cada *topic* (a la derecha).

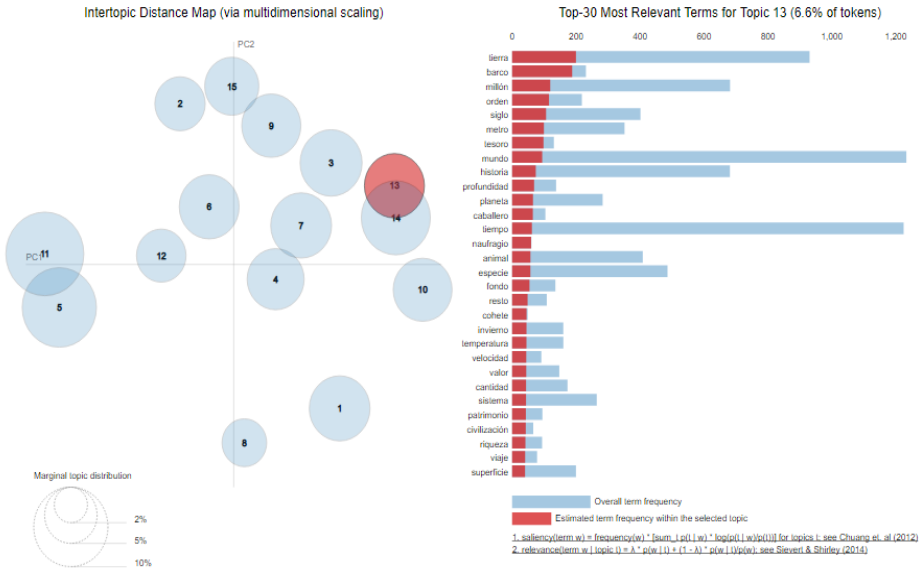


Fig. 3.2: Representación gráfica en pyLDAvis de los resultados del TM basado en LDA.

3.6. Sistema conjunto

Una vez obtenidas las entidades identificadas mediante Apache OpenNLP en un texto, y una vez realizado el modelado de *topics* a través de *LDA* para el corpus de documentos, se plantea el siguiente sistema que interrelaciona los resultados obtenidos de uno y otro procedimiento. Los pasos seguidos son los siguientes:

1. En primer lugar, se pre-procesa el texto, para adaptar el formato al sistema de modelado de *topics* que se emplea posteriormente. Se eliminan los signos de puntuación del texto, y todas las palabras se pasan a minúsculas. Para no perder la referencia a las entidades reconocidas por R en el apartado de *NER*, se sustituyen las etiquetas desde el formato:

   "<START:person>Federer</END>"

   que era generado en el código R, al nuevo formato:

   "entidad_person_Federer"

2. A continuación, se almacena qué entidades se han conservado en el diccionario creado para el modelo de *topics*, obteniendo sus id. El diccionario crea un recuento de todas las palabras que aparecen en el corpus de documentos, filtrando después los verbos, preposiciones, adverbios o adjetivos, junto con las palabras que tienen índices de aparición muy bajos (son tan infrecuentes que no caracterizan la temática) o demasiado altos (palabras tan comunes que no son propias del *topic*).

3. A partir de esos id obtenidos se comprueba la probabilidad de las entidades existentes para cada *topic*. Esto permite identificar el tópico o los tópicos para los cuales una entidad es más significativa (en qué tópico aparece más dicha entidad), y ordenar los tópicos en función de su importancia para una entidad, o viceversa.

4. Se puede obtener también un vector de probabilidades de aparición para uno y otro tópico de la entidad.

5. A partir del vector de probabilidades obtenido en el paso 3, se puede llevar a cabo una comparación entre entidades, obteniendo una medida de similitud entre sus vectores. Así se puede descubrir qué entidades poseen un comportamiento similar en cuanto a sus apariciones dentro de una temática.

4. RESULTADOS Y EVALUACIÓN

4.1. Medidas de evaluación

Para estudiar la calidad de los sistemas que se han implementado se han desarrollado y evaluado por separado las distintas partes de este proyecto.

4.1.1. *NER*

Un sistema *NER* de identificación de un tipo de entidades es un problema de detección, o clasificación en dos clases (decisión). Cada una de las palabras de un texto corresponde o no a una entidad, y el decisor debe asignar a cada palabra la categoría "entidad" (salida positiva) o la categoría "no-entidad" (salida negativa).

Por ello, el acierto en el reconocimiento de las entidades de un texto es un número cuantificable, que relacionará el número de entidades correctamente etiquetadas, con el número de las que se etiquetaron sin ser entidades, y entidades que no se etiquetaron.

Los resultados para las dos tecnologías se expresan en una tabla como la siguiente:

Tabla 4.1: En verde, palabras correctamente clasificadas (entidad/no-entidad) y en rojo, incorrectamente.

Entidades	**Etiquetadas:**	**No etiquetadas:**
Reales:	VP: entidades etiquetadas	FN: entidades no etiquetadas
Falsas:	FP: no-entidades etiquetadas	VN: no-entidades no etiquetadas

En verde aparecen los casos de **acierto**:

- **Verdadero Positivo** son aquellos en que las muestras (en este caso, palabras) corresponden a la clase que se quiere detectar (entidad), y además son etiquetados.

- Igualmente, los casos **Verdadero Negativo** son aquellas muestras que no pertenecen a la clase y no se etiquetan como si pertenecieran.

En color rojo aparecen los casos de **error**, en que las muestras han sido tratadas incorrectamente:

- **Falso Negativo** son los casos en que la muestra pertenece a la clase, pero el sistema la clasifica como muestra negativa (no perteneciente). En el ámbito de la detección, esto se conoce como **pérdida**, puesto que un caso real ha sido ignorado como si no lo fuese.

- **Falso Positivo** son los casos en que la muestra no pertenece, pero se etiqueta como positivo (perteneciente a la clase). En el ámbito de la detección se conoce como **falsa alarma**, debido al origen de estos sistemas para identificación radar de

aviones enemigos durante la Segunda Guerra Mundial, en que algo era detectado como avión sin serlo realmente.

Utilizando estos valores, se calculan las métricas de error:

Tabla 4.2: Métricas de error.

Accuracy	$\frac{VP+VN}{VP+VN+FN+FP}$	"Exactitud", porcentaje de aciertos sobre el total de casos
Precision	$\frac{VP}{VP+FP}$	"Precisión", aciertos positivos frente a positivos (falsos y verdaderos)
Recall, Sensitivity	$\frac{VP}{VP+FN}$	"Sensibilidad", aciertos positivos sobre los que deberían haber sido (detectados y no detectados). También "*True Positive Rate"*
Specifity	$\frac{VN}{VN+FP}$	"Especifidad", "no-entidades" correctamente no etiquetadas sobre las que deberían haber sido. También "*True Negative Rate*"

Ninguna de estas métricas es suficiente por sí sola, sino que para unos buenos resultados debe buscarse un compromiso entre ellas. Por ejemplo, para un *recall* o sensibilidad (porcentaje de positivos encontrados sobre el total de positivos existentes) del 100%, bastaría con que el sistema etiquetase todos los posibles casos como positivos, y así jamás perdería un caso positivo, pero evidentemente este sistema sería absurdo.

Para relacionar estos dos parámetros, se calcula la métrica *F1 score*, que es la media armónica que los relaciona:

$$F1 = 2\frac{Precision*Recall}{Precision+Recall} \qquad (4.1)$$

Cuanto más próximo a 1 sea este parámetro, más equilibrados y cercanos a 1 serán los valores de *precisión* y *recall*, mientras que si uno de los dos tiene un valor muy bajo, decrece drásticamente el valor de *F1*.

4.1.2. *TM*

El caso de *TM* es completamente distinto. Por ser una tecnología no supervisada no existe una única solución, ni la solución perfecta, ni unos patrones cognoscibles a identificar. La asignación de temáticas a un texto puede ser muy variada sin dejar de ser correcta, y no existe un método cuantitativo para la valoración de los resultados de forma estandarizada.

Sin existir una medida del error o acierto de los *topics* entra en juego el criterio del usuario para aceptarlos como razonables o descartarlos. El parámetro más calculado computacionalmente para medir la convergencia del aprendizaje de los *topics* es la **perplejidad** o ***perplexity***, que caracteriza la evolución en el entrenamiento de un modelo

de *topics* hasta que se estabiliza, para encontrar así el punto óptimo en que dejar de entrenar.

$$perplexity\ (D) = \exp(-\frac{\sum_{d=1}^{M} \log p\ (w_d)}{\sum_{d=1}^{M} N_d}) \qquad (4.2)$$

Donde D es el corpus en el que se evalúa, M es el número de documentos, $p(w_d)$ verosimilitud del documento, aproximada a través de las distribuciones aprendidas, y N_d el número de palabras de cada documento. [12]

Este parámetro se obtiene a partir de la estimación de la verosimilitud (*likelihood*), según la siguiente fórmula:

$$perplexity = 2^{-\ln(Likelihood)} \qquad (4.3)$$

La **verosimilitud** no puede obtenerse de forma exacta, sino que se aproxima a través de la cota superior. Si quisiésemos volver a obtener la verosimilitud desde la perplejidad, despejaríamos en la ecuación:

$$LogLikelihood = 2^{-perplexity} \qquad (4.4)$$

$$Likelihood = e^{-log_2\ perplexity} \qquad (4.5)$$

Sin embargo, estos parámetros (*perplexity*, log-verosimilitud) no son suficientes en sí mismos para caracterizar la calidad de los *topics* obtenidos. Se muestra a continuación (Fig. 4.5) la comparativa entre dos ejemplos de curvas de perplejidad, y cómo, pese a la diferencia entre los valores de una y otra, al haber alcanzado su convergencia, humanamente se perciben como igual de buenos.

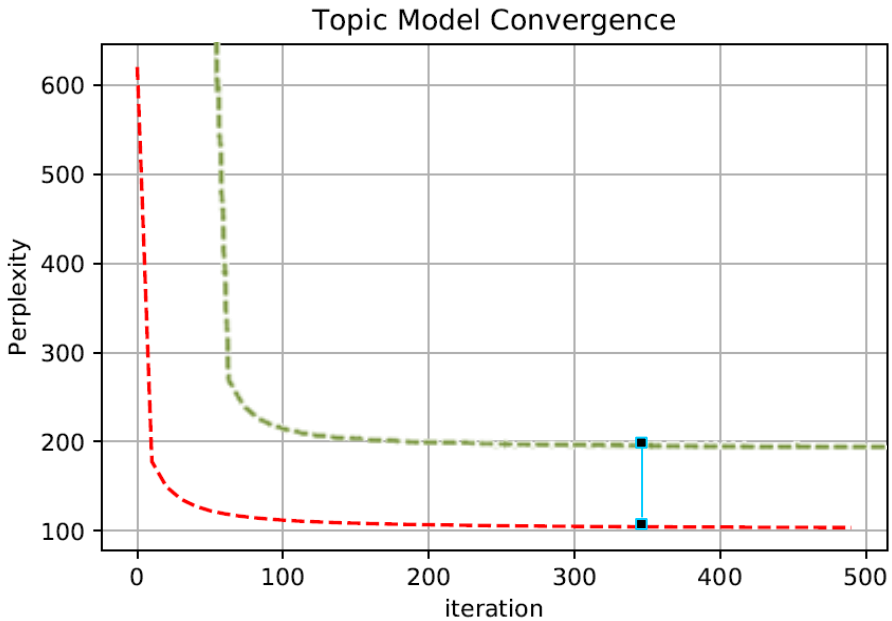


Fig. 4.1: Ejemplo de dos curvas de perplejidad que en la evaluación humana generaban la misma respuesta, sin importar la distancia entre las dos curvas

Se introducen estudios sobre grupos de personas (***Human Task***) que evalúen la calidad de los tópicos obtenidos [34]. Uno de los estudios más realizados consiste en pruebas sobre la coherencia de los términos que caracterizan un *topic*, empleando para ello a un conjunto de personas. Una prueba que se realiza con frecuencia es aquella que consiste en sustituir aleatoriamente un término dentro de un *topic*, y solicitar identificar el término intruso. Por ejemplo:

Topic 1 {“gato”, “arena”, “radio”, “limpiar”, “sofá”, “lámpara”}

quedaría sustituido por:

*Topic 1** {“gato”, “arena”, “radio”, “Toledo”, “sofá”, “lámpara”}

En ocasiones, estos intrusos son fáciles de identificar, pero en otros casos en que los términos no están tan relacionados, puede suceder que el intruso no destaque frente a los otros términos. Cuanto más relacionados estén entre sí los términos, más información proporcionan y mejor será el *topic*. Esta prueba se conoce como **prueba de intrusión**.

Paralelamente a los estudios con sujetos existen investigaciones tratando de obtener métricas de coherencia que proporcionen resultados similares a los que daría un ser humano, aunque aún los mejores resultados no llegan al 80% de acierto sobre las respuestas humanas (Fig. 4.2)

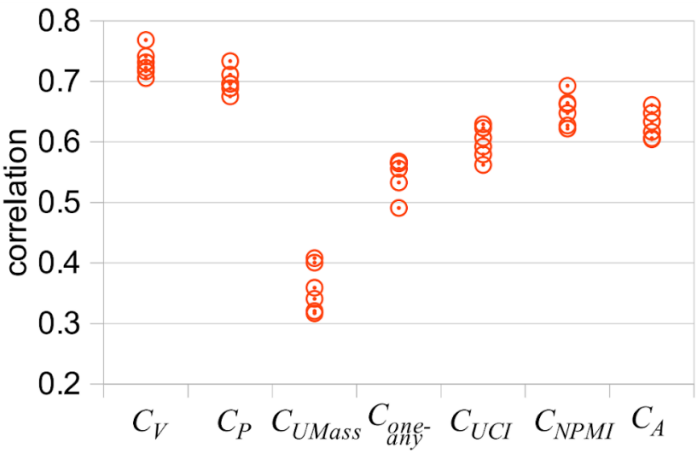


Fig. 4.2: Resultados para distintas medidas de coherencia estudiados frente a la respuesta humana. Fuente: Röder [35]

Por ello, se propone una nueva familia de medidas para evaluar la calidad de estos modelos de *topics*: la **coherencia** [35], que se aplica a las “n” palabras principales dentro de un *topic*. Típicamente se define como el promedio de las medidas de similitud de las palabras de un documento, obtenidas por pares (cada palabra con todas las demás). Aquellos modelos que tengan unos *topics* más relacionados, tendrán unas medidas de coherencia más altas.

4.1.3. Sistema conjunto

El sistema conjunto de procesamiento de texto que se ha desarrollado consta de un módulo *NER* y otro *TM*, cuyos resultados se complementan para proporcionar una información más completa de la información contenida en el texto.

Para evaluar la relación entre los resultados de ambos procesos, se propone caracterizar las entidades con sus probabilidades de pertenencia a cada uno de los tópicos, para así poder calcular cómo de similares son en cuanto a las temáticas en las que aparecen. De esta manera, se pueden obtener medidas de similitud de entidades. En este proyecto se emplea la métrica de similitud llamada "**similitud coseno**" ("*Cosine similarity*"). Esta medida se basa en el producto escalar entre dos vectores, que quedan proyectados sobre una única dirección a través del coseno del ángulo que forman. Así, dos vectores en la misma dirección tendrán un producto escalar máximo e igual al valor del producto de sus módulos, mientras que dos vectores perpendiculares entre sí formarán un ángulo de 90º, cuyo coseno es nulo, y tendrán similitud 0 por ser perpendiculares.

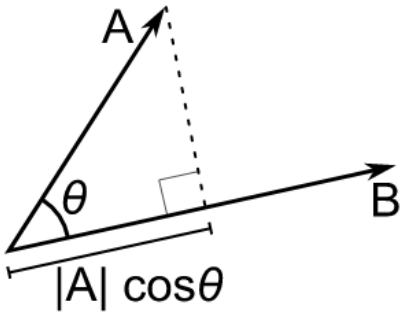


Fig. 4.3: Ejemplo de la proyección por el producto escalar del vector A sobre la dirección del vector B. Debido a la diferencia en la dirección (ángulo θ ≈ 30º), el efecto de A sobre B (su proyección) es menor respecto a la magnitud de A. Fuente: Wikipedia [36].

Matemáticamente, se obtiene así:

$$similitud\ coseno = \cos(\theta) = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2} \cdot \sqrt{\sum_{i=1}^{n} {B_i}^2}} \qquad (4.6)$$

siendo A_i y B_i componentes de los vectores A y B, respectivamente.

Existen otras formas de obtener la similitud, pero en este sistema es la similitud coseno la empleada como medida de similitud.

4.2. Resultados y evaluación

A continuación, se presentan las tablas de resultados y se evalúan los parámetros explicados en el apartado anterior.

4.2.1. *NER*

Para el fin comparativo entre las tecnologías estudiadas se procesaron los mismos archivos a través de una implementación *NLP* en Java y en R. Las entidades identificadas y perdidas en este proceso fueron las siguientes:

Tabla 4.3: Comparativa entre los resultados NER en R y en Java

	R	**Java**
Entidades etiquetadas correctamente	88 correctas	67 correctas
Entidades aprox. correctas	~8 no exactas	~12 no exactas
Entidades reales	160 entidades reales	160 entidades reales
Entidades etiquetadas sin serlo	6, 8.3% de las entidades detectadas no corresponden	10, 15.2% de las entidades detectadas no corresponden
Total etiquetadas	102 etiquetadas	89 etiquetadas

A partir de estos datos, se puede obtener una matriz de confusión del error cometido desde cada implementación (ver Tabla 4.1 para la explicación de los campos):

Tabla 4.4: Entidades evaluadas en la implementación R

Entidades	**Etiquetadas**	**No etiquetadas**	**Total:**
Reales	~ 96 (contando las aprox)	64 no encontradas	160 entidades
Falsas	6	8.625 palabras	8.631 “no-entidades”
Total:	102 etiquetadas	8.689 ignoradas	8.791 palabras

Tabla 4.5: Entidades evaluadas en la implementación Java.

Entidades	**Etiquetadas**	**No etiquetadas**	**Total:**
Reales	~ 79 (contando las aprox)	81 no encontradas	160 entidades
Falsas	10	8.621 palabras	8.631 “no-entidades”
Total:	89 etiquetadas	8.702 ignoradas	8.791 palabras

Las medidas del error (ver Tabla 4.2 para las descripciones de cada parámetro) de ambas tecnologías:

Tabla 4.6: Métricas calculadas sobre las entidades evaluadas en R y en Java. Se puede observar que el procesamiento en R proporciona mejores estadísticos para todos los casos, frente a la implementación R.

	Expresión	R	Java
Accuracy	$\frac{VP + VN}{VP + VN + FN + FP}$	$\frac{96 + 8625}{8791} = 99{,}20\%$	$\frac{79 + 8625}{8791} = 99{,}01\%$
Precision	$\frac{VP}{VP + FP}$	$\frac{96}{96 + 6} = 94{,}12\%$	$\frac{79}{79 + 10} = 88{,}76\%$
Recall, Sensitivity	$\frac{VP}{VP + FN}$	$\frac{96}{96 + 64} = 60\%$	$\frac{79}{79 + 81} = 49{,}37\%$
Specifity	$\frac{VN}{VN + FP}$	$\frac{8625}{8625 + 6} = 99{,}93\%$	$\frac{8621}{8621 + 10} = 99{,}88\%$
F1 score	$2\frac{Precision * Recall}{Precision + Recall}$	$2\frac{94{,}12 * 60}{94{,}12 + 60} = 73{,}28\%$	$2\frac{88{,}76 * 49{,}37}{88{,}76 + 49{,}37} = 63{,}45\%$

Como se puede observar en la tabla, el procesamiento a través de R ha proporcionado unos mejores resultados, puesto que ha identificado correctamente más entidades y no ha etiquetado tantas "no-entidades" como en Java. Resulta inesperado, puesto que ambos sistemas empleaban como base las librerías de procesamiento de lenguaje natural desarrolladas por Apache OpenNLP, y se ha buscado el mayor paralelismo posible entre ambos sistemas para poder evaluar los resultados.

Durante el reconocimiento de entidades, el texto de los archivos de subtitulado de RTVE ha sido etiquetado en las entidades identificadas. Se muestra un ejemplo a continuación:

> *Es la fragata Méndez Núñez, el buque que España ha retirado del grupo de combate de Estados Unidos camino del golfo Pérsico, ante la escalada de tensión con Irán. Participaba en un ejercicio de entrenamiento junto al portaaviones estadounidense Abraham Lincoln. Hace 10 días, Washington decidió enviar esa flota al Golfo para responder a un posible ataque de Irán a intereses norteamericanos. Buenas tardes. El gobierno resta importancia a la retirada de la fragata y la justifica porque dice que Estados Unidos cambió la misión prevista, acordada hace más un año. Aunque insiste en que es una retirada provisional. Ahora el gobierno de Estados Unidos ha marcado una misión; una misión que no estaba prevista con el gobierno español. Y desde ese punto de vista, nosotros temporalmente interrumpimos ese acompañamiento, esa interacción. El Supremo autoriza que los presos del procés electos asistan el día 21, en el Congreso, a la constitución de las nuevas Cortes. El Tribunal rechaza suspender el juicio, como habían pedido las defensas, y tampoco pedirá permiso a las cámaras para seguir juzgándoles. El acuerdo para la mesa del Congreso se da casi por cerrado. PSOE y Unidas Podemos hablan de buenas sensaciones. En la mesa no estarían ni los grupos nacionalistas ni VOX. Los diputados siguen recogiendo sus actas. Entre los que han ido hoy, el presidente de Vox, Santiago Abascal, o los representantes del PNV.* (…)

Este fragmento corresponde al comienzo de un telediario del día 13 de mayo de 2019, que el sistema *NER* de Apache OpenNLP en R etiqueta de la siguiente manera:

> *Es la fragata Méndez Núñez, el buque que España ha retirado del grupo de combate de <START:organization>Estados Unidos<END> camino del golfo Pérsico, ante la escalada de tensión con Irán. Participaba en un ejercicio de entrenamiento junto al portaaviones estadounidense <START:person>Abraham Lincoln<END>. Hace 10 días, Washington decidió enviar esa flota al Golfo para responder a un posible ataque de Irán a intereses norteamericanos. Buenas tardes. El gobierno resta importancia a la retirada de la fragata y la justifica porque dice que <START:organization>Estados Unidos<END> cambió la misión prevista, acordada hace más un año. Aunque insiste en que es una retirada provisional. Ahora el gobierno de <START:organization>Estados Unidos<END> ha marcado una misión; una misión que no estaba prevista con el gobierno español. Y desde ese punto de vista, nosotros temporalmente interrumpimos ese acompañamiento, esa interacción. El <START:organization>Supremo<END> autoriza que los presos del procés electos asistan el día 21, en el <START:organization>Congreso<END>, a la constitución de las nuevas <START:organization>Cortes<END>. El <START:organization>Tribunal<END> rechaza suspender el juicio, como habían pedido las defensas, y tampoco pedirá permiso a las cámaras para seguir juzgándoles. El acuerdo para la mesa del <START:organization>Congreso<END> se da casi por cerrado. <START:organization>PSOE<END> y Unidas Podemos hablan de buenas sensaciones. En la mesa no estarían ni los grupos nacionalistas ni <START:organization>VOX<END>. Los diputados siguen recogiendo sus actas. Entre los que han ido hoy, el presidente de Vox, Santiago Abascal, o los representantes del <START:organization>PNV<END>* (…)

Para el modelado de *topics* en Python, el etiquetado de entidades se modifica para no ser alterado por el procesamiento posterior del texto (ver siguiente fragmento). A partir del reconocimiento de las entidades en el texto se elabora una lista de las mismas.

> *es la fragata méndez núñez, el buque que españa ha retirado del grupo de combate de entidad_organization_estados unidos camino del golfo pérsico, ante la escalada de tensión con irán. participaba en un ejercicio de entrenamiento junto al portaaviones estadounidense entidad_person_abraham lincoln. hace 10 días, washington decidió enviar esa flota al golfo para responder a un posible ataque de irán a intereses norteamericanos. buenas tardes. el gobierno resta importancia a la retirada de la fragata y la justifica porque dice que entidad_organization_estados unidos cambió la misión prevista, acordada hace másun año. aunque insiste en que es una retirada provisional. ahora el gobierno de entidad_organizationestados unidos ha marcado una misión; una misión que no estaba prevista con el gobierno español. y desde ese punto de vista, nosotros temporalmente interrumpimos ese acompañamiento, esa interacción. el entidad_organization_supremo autoriza que los presos del procés electos asistan el día 21, en el entidad_organization_congreso, a la constitución de las nuevas entidad_organization_cortes. el entidad_organization_tribunal rechaza suspender el juicio, como habían pedido las defensas, y tampoco pedirá permiso a las cámaras para seguir juzgándoles. el acuerdo para la mesa del entidad_organization_congreso se da casi por cerrado. entidad_organization_psoe y unidas podemos hablan de buenas sensaciones. en la mesa no estarían ni los grupos nacionalistas ni entidad_organization_vox. los diputados siguen recogiendo sus actas. entre los que han ido hoy, el presidente de vox, santiago abascal, o los representantes del entidad_organization_pnv (…)*

4.2.2. *TM*

Como se ha explicado en el apartado 4.1.2, los modelos de *topics* no tienen una métrica de error. La valoración de sus resultados se realiza a través de inspección visual humana (pruebas de intrusión de términos extraños dentro de un *topic* para que un conjunto de personas identifique el que no pertenecía a la lista original) o con aproximaciones matemáticas, que no llegan a predecir con más de 70% de acierto la respuesta humana.

Entre los parámetros estadísticos más empleados se encuentra la **perplejidad**, o ***perplexity***, que es el resultado al que converge el modelo, según lo explicado previamente. Este parámetro se basa en la estimación de la log-verosimilitud, que, al no poder calcularse de forma exacta, se aproxima con una cota, y no es una tasa de error.

A partir del modelo de *topics* obtenido en este sistema, se ha representado gráficamente la perplejidad y la log-verosimilitud, para encontrar el punto de convergencia del modelo:

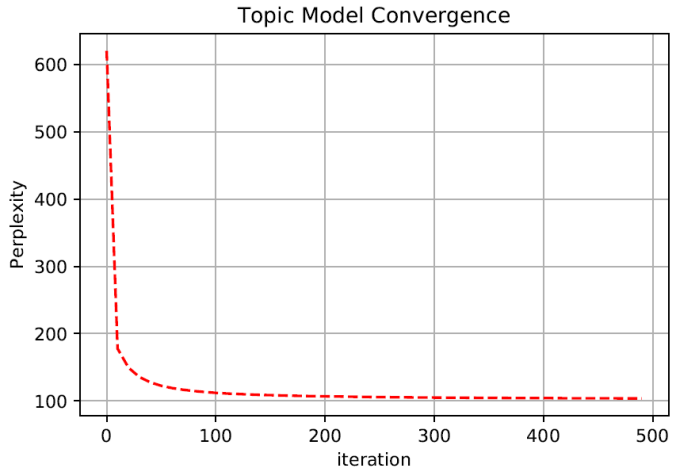


Fig. 4.4: Curva de evolución de la perplejidad con el número de iteraciones en el entrenamiento para el modelo desarrollado en este trabajo.

En la Fig. 4.3 se muestra la perplejidad obtenida sobre el modelo de *topics* entrenado para este proyecto. Se puede observar cómo se estabiliza su valor en torno a 250 iteraciones del proceso, que será el número óptimo de iteraciones para un modelo lo más coherente posible.

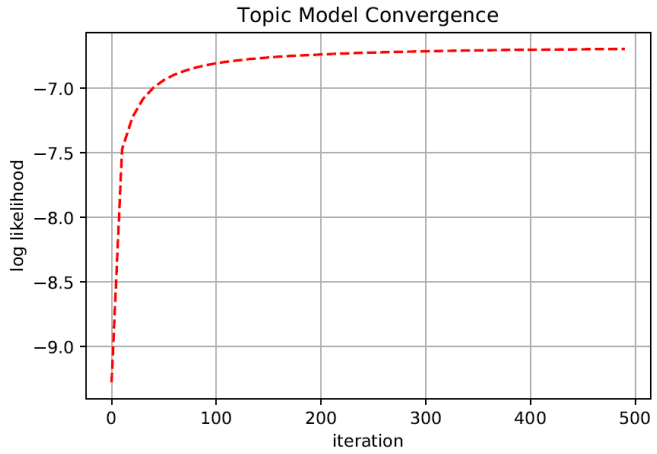


Fig. 4.5: Curva de evolución de la log-verosimilitud con el número de iteraciones en el entrenamiento para el modelo desarrollado en este trabajo.

En la Fig. 4.4 se muestra el parámetro log-verosimilitud, en el que se basa la perplejidad, según las fórmulas expuestas en el apartado 4.1.2. De nuevo, puede verificarse también en este caso cómo en torno a 250 iteraciones el sistema se estabiliza.

Por lo expuesto en dicho apartado, no resulta suficiente con medir estos parámetros para conocer la calidad de un *topic*. Por esto, se evalúan también los resultados de **coherencia** sobre el modelo:

Tabla 4.7: Valores de coherencia NPMI obtenidos para los tópicos de este modelo, ordenados de mejor a peor coherencia.

Nº	**Coherencia (npmi)**	**Términos del *topic***
1	0.0025630357621744915	guerra, personaje, alemán, final, película, historia, ajedrez, puerta, momento, verdad
2	0.0004802325361501071	tiempo, planeta, energía, ciudad, mundo, tierra, edificio, especie, temperatura, aspecto
3	-0.008555883666803358	tierra, bosque, animal, hembra, neandertal, especie, preso, entidad_location_europa, costa, territorio
4	-0.016220985268013192	jenny, entidad_person_vincent, músico, reginald, chico, mundo, gracia, entidad_person_ah, entidad_location_barcelona, noche
5	-0.02423684608626974	energía, central, reactor, empresa, político, elección, ciudadano, gente, problema, situación
6	-0.024278432688024962	mundo, millón, cerebro, historia, barco, siglo, tecnología, hombre, realidad, mente
7	0.061949057563225364	hombre, gracia, señor, verdad, músico, tiempo, padre, entidad_organization_ríe, favor, entidad_person_ah
8	-0.06302949425612302	millón, continente, gracia, entidad_location_australia, desierto, tierra, historia, mundo, fruto, túnel
9	0.08591470979228039	madre, entidad_person_antonio, verdad, padre, entidad_person_carlos, coche, favor, hombre, dinero, familia
10	-0.10568048732741589	entidad_person_el, entidad_location_madrid, campeón, derecho, semana, final, españa, jugador, momento, victoria
11	-0.12529461226706612	novio, caballo, familia, verdad, brazo, tierra, gente, sabor, carne, hotel
12	-0.1253963832271048	gracia, entidad_person_daniel, entidad_organization_grita, serge, mundo, músico, tabla, entidad_organization_llora, guion, entidad_location_caldeya
13	-0.1316515746291053	película, entidad_person_segunda, entidad_location_españa, tiempo, director, rodaje, actor, noche, momento, entidad_person_john
14	-0.16435155330147563	entidad_person_enrique, esposo, catalina, orden, caballero, hombre, reunión, entidad_person_ana, entidad_location_europa, silbato
15	-0.4320333282652562	célula, castaño, entidad_organization_adn, harina, molécula, envejecimiento, combinación, entidad_location_córcega, pizarra, creps

Se puede apreciar cómo los tópicos presentados en las primeras filas de la tabla:

[guerra, personaje, alemán, final, película, historia, ajedrez, puerta, momento, verdad]
[tiempo, planeta, energía, ciudad, mundo, tierra, edificio, especie, temperatura, aspecto]

[tierra, bosque, animal, hembra, neandertal, especie, preso, entidad_location_europa, costa, territorio]
…
[energía, central, reactor, empresa, político, elección, ciudadano, gente, problema, situación]
[mundo, millón, cerebro, historia, barco, siglo, tecnología, hombre, realidad, mente]

pertenecen a un ámbito más concreto que otros de menor coherencia, que agrupan términos más dispares:

[película, entidad_person_segunda, entidad_location_españa, tiempo, director, rodaje, actor, noche, momento, entidad_person_john]
[entidad_person_enrique, esposo, catalina, orden, caballero, hombre, reunión, entidad_person_ana, entidad_location_europa, silbato]
[célula, castaño, entidad_organization_adn, harina, molécula, envejecimiento, combinación, entidad_location_córcega, pizarra, creps]

Como se muestra en la Fig. 4.2, existen distintas medidas de coherencia aplicables sobre un modelo de *topics*. En la Tabla 4.7 se ha empleado la medida "NPMI". Los resultados obtenidos tanto con esta como con otras de las medidas son muy bajos y muy similares para todos los tópicos que se evalúan, debido a que estos valores de coherencia están optimizados para modelos en inglés, y entrenados en corpus de documentos externos, por ejemplo, a partir de artículos de Wikipedia. Se obtendrían mejores resultados desarrollando medidas de coherencia entrenadas dentro de un corpus de gran cantidad de documentos de RTVE, y optimizando estas medidas para trabajar con documentos en español.

El modelado de *topics* busca obtener las temáticas principales que aparecen en los documentos. Realizando la búsqueda inversa, para un *topic* entre los existentes, se obtiene el documento en el que esa temática es mayoritaria. Se muestra como ejemplo la temática que agrupa estas palabras:

['barco', 'edificio', 'fruto', 'metro', 'madre', 'hembra', 'hormigón', 'cabaña', 'tiempo', 'piedra']

Para la que el documento en que tiene mayor importancia ese *topic* pertenece a un documental de la serie "Grandes Diseños", sobre construcciones[10], que efectivamente, trata de un tema relacionado con esos términos:

(…) sí. pero la nueva perspectiva de entidad_person_Fred y entidad_person_Saffron durará poco. un repentino frío anuncia el comienzo del invierno, y el último vertido de hormigón para los techos se ha retrasado casi un mes. en diciembre, la familia tuvo que mudarse con su nueva casera ahora mismo están mi hijo, su mujer y sus dos hijos viviendo conmigo. ahora los míos. ya están, sí. ya los he limpiado. los limpió anoche. sí, lo hice anoche. he aprendido a morderme la lengua. nunca pondría una botella de leche en la mesa ni en el desayuno. hoy se sigue con los tejados. nos ha llevado demasiado tiempo llegar a

[10] Documental "Grandes Diseños", temporada 15, episodio 7, de emisión en enero de 2018. Disponible en: http://www.rtve.es/alacarta/videos/otros-documentales/otros-documentales-grandes-disenos-15-episodio-7/4434977/

este punto, pero ya estamos aquí, no está nevando, y eso es algo bueno. llevamos seis meses de retraso, principalmente por el trabajo preliminar. de hecho, es en su totalidad por el trabajo preliminar. este será el último día del equipo de trabajo preliminar. un trabajo que se suponía que tardaría tres meses, ha tardado cerca de siete sí, ha sido un trabajo difícil. el acceso ha sido complicado. hemos revisado el programa, pero la calidad no se mide con la rapidez. ya en el año siguiente, el granero del piso superior ya está, y la primera pieza de madera se abre paso. este debería ser un día trascendental, pero entidad_person_Saffron tuvo malas noticias de parte de los contratistas. no va a estar terminado hasta junio. he tardado dos meses en aceptarlo, creo. es difícil de sobrellevar; me dejó sin aliento. esperaba mudarme de nuestro piso alquilado, y que con eso sentiría una energía renovada. para este proyecto, pero eso no sucedió. lo que sí sucedió fue. que el estrés me invadió, y realmente. no pude hacer mucho. la dificultad de entidad_person_Saffron viene de la mano de un duro invierno. pero en abril, ella y entidad_person_Fred han reestructurado sus finanzas. (…)

El fragmento que se muestra aparece etiquetado en entidades reconocidas en R en el bloque previo ("*entidad_person_Fred*", "*entidad_person_Saffron*"), identificadas para poder ser recuperadas.

Por último, se muestran los resultados gráficos del modelo de *topics* entrenado, mediante la visualización empleando la librería pyLDAvis:

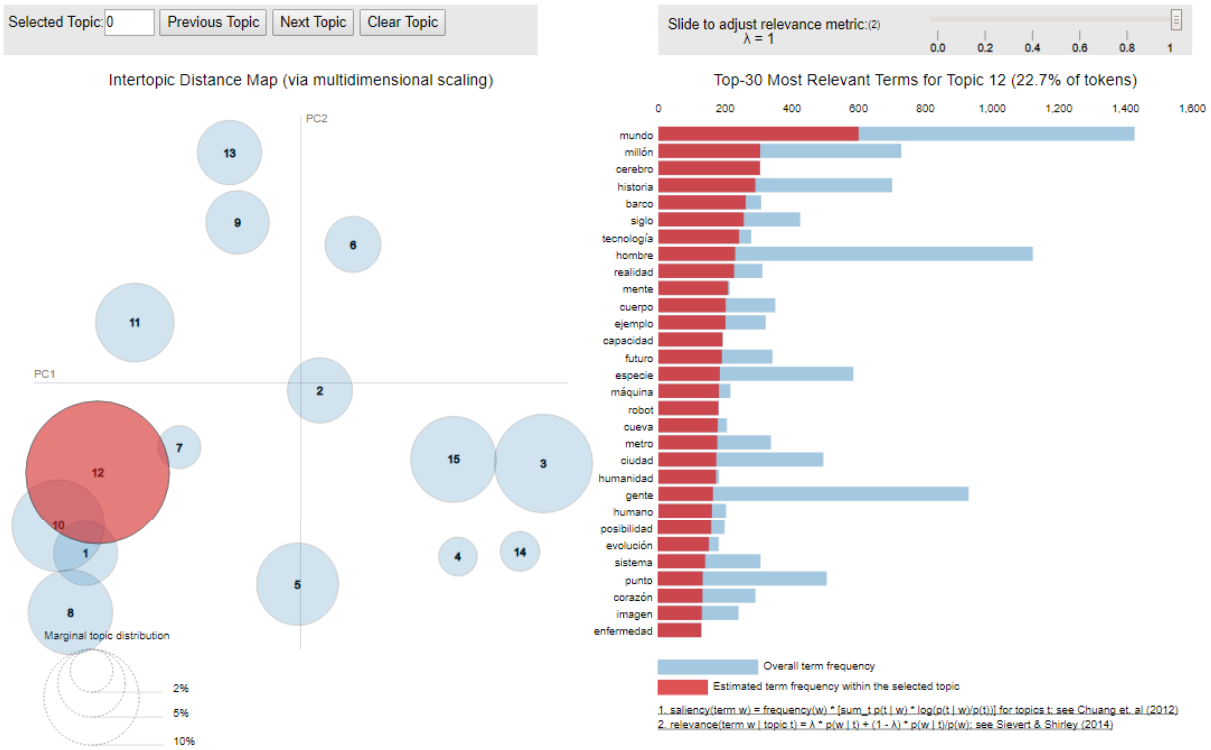


Fig. 4.6: A la izquierda, burbujas que representan los tópicos obtenidos. Se ha seleccionado el topic 12 (en rojo), de forma que a la derecha aparecen las probabilidades de los 30 términos mayoritarios dentro del topic para ese documento (rojo), frente al promedio general (azul)

4.2.3. Sistema conjunto

Para la evaluación de la calidad del sistema conjunto se han estudiado las relaciones entre entidades encontradas y *topics* obtenidos. Como ejemplo, se selecciona la entidad:

"entidad_location_europa"

que ha sido reconocida en el corpus de documentos, y se recupera el *topic* más similar a ella, en el que tiene mayor peso esa entidad:

['tierra', 'bosque', 'animal', 'hembra', 'neandertal', 'especie', 'preso',
'entidad_location_europa', 'costa', 'territorio']

Se ha creado una matriz que relaciona todas las entidades frente a los *topics* en que aparecen, de la siguiente manera (Tabla 4.8). En la primera fila se muestran las probabilidades de aparición de una cierta entidad ("entidad_location_europa") dentro de cada uno de los *topics* con los que se trabaja en este sistema. Se observa que la distribución de esta entidad se da mayoritariamente en cuatro *topics*:

Tabla 4.8: Aparición de la entidad seleccionada dentro de algunos topics del modelo, junto con la probabilidad de aparición en cada uno de ellos.

Probabilidad	**Términos característicos del topic**	**Nº**	**Etiqueta**
0.49350444	tierra, bosque, animal, hembra, neandertal, especie, preso, entidad_location_europa, costa, territorio	7	Naturaleza, evolución
0.47478195	entidad_person_enrique, esposo, catalina, orden, caballero, hombre, reunión, entidad_person_ana, entidad_location_europa, silbato	1	Personajes históricos
0.02400231	tiempo, planeta, energía, ciudad, mundo, tierra, edificio, especie, temperatura, aspecto	9	Clima, planeta
0.0077113	mundo, millón, cerebro, historia, barco, siglo, tecnología, hombre, realidad, mente	11	Historia, mundo
0	(otras temáticas obtenidas en el modelo de tópicos)	…	…
Total Σ = 1			

Se puede comprobar cómo los *topics* en los que es más significativa esa entidad guardan relación directa con el significado de esta entidad. A partir de estas probabilidades se pueden recuperar otras entidades con una distribución de probabilidad similar a la de la entidad que se estudia. En la Tabla 4.8, las siguientes entidades con una distribución más parecida se muestran como filas bajo la de la entidad "entidad_location_europa"

Tabla 4.9: Para la entidad de tipo lugar “Europa”, la distribución de importancia en apariciones en las temáticas aparece en la primera fila. Justo debajo aparecen las 10 entidades con una distribución de aparición en los topics más similar.

Términos que caracterizan los 15 *topics* obtenidos para este corpus de documentos	millón, continente, gracia, entidad_location_australia, desierto, tierra, historia, mundo, fruto, túnel	entidad_person_enrique, esposo, catalina, orden, caballero, hombre, reunión, entidad_person_ana, entidad_location_europa, silbato	hombre, gracia, señor, verdad, músico, tiempo, padre, entidad_organization_ríe, favor, entidad_person_ah	gracia, entidad_person_daniel, entidad_organization_grita, serge, mundo, músico, tabla, entidad_organization_llora, guion, entidad_location_caldeya	novio, caballo, familia, verdad, brazo, tierra, gente, sabor, carne, hotel	guerra, personaje, alemán, final, película, historia, ajedrez, puerta, momento, verdad	célula, castaño, entidad_organization_adn, harina, molécula, envejecimiento, combinación, entidad_location_córcega, pizarra, creps	tierra, bosque, animal, hembra, neandertal, especie, preso, entidad_location_europa, costa, territorio	entidad_person_el, entidad_location_madrid, campeón, derecho, semana, final, españa, jugador, momento, victoria	tiempo, planeta, energía, ciudad, mundo, tierra, edificio, especie, temperatura, aspecto	energía, central, reactor, empresa, político, elección, ciudadano, gente, problema, situación	mundo, millón, cerebro, historia, barco, siglo, tecnología, hombre, realidad, mente	película, entidad_person_segunda, entidad_location_españa, tiempo, director, rodaje, actor, noche, momento, entidad_person_john	jenny, entidad_person_vincent, músico, reginald, chico, mundo, gracia, entidad_person_ah, entidad_location_barcelona, noche	madre, entidad_person_antonio, verdad, padre, entidad_person_carlos, coche, favor, hombre, dinero, familia
Identificación topic	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
entidad_ location_ europa	0.	0.47478195	0.	0.	0.	0.	0.	0.49350444	0.	0.02400231	0.	0.0077113	0.	0.	0.
Las entidades más semejantes a ella en importancia para estos *topics* son las siguientes:															
entidad_ person_ república	0.	0.	0.	0.	0.	0.	0.	0.97490989	0.	0.0259011	0.	0.	0.	0.	0.
entidad_ person_ cárpatos	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.
entidad_ person_ urss	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.
entidad_ location_ polonia	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.
entidad_organization_occidente	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.
entidad_location_suecia	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.
entidad_location_skagerrak	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.
entidad_location_rumanía	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.
entidad_location_santa	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.
entidad_location_eslovaquia	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.
entidad_organization_naciones	0.	0.	0.	0.	0.09125291	0.	0.	0.90874709	0.	0.	0.	0.	0.	0.	0.
entidad_location_puente	0.	0.	0.	0.	0.	0.	0.	0.83108	0.	0.16892	0.	0.	0.	0.	0.
entidad_location_austria	0.	0.	0.	0.	0.	0.	0.	0.79321047	0.	0.	0.	0.	0.20678953	0.	0.
entidad_person_antoine	0.	0.96863203	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.03136797	0.	0.	0.

5. ORGANIZACIÓN

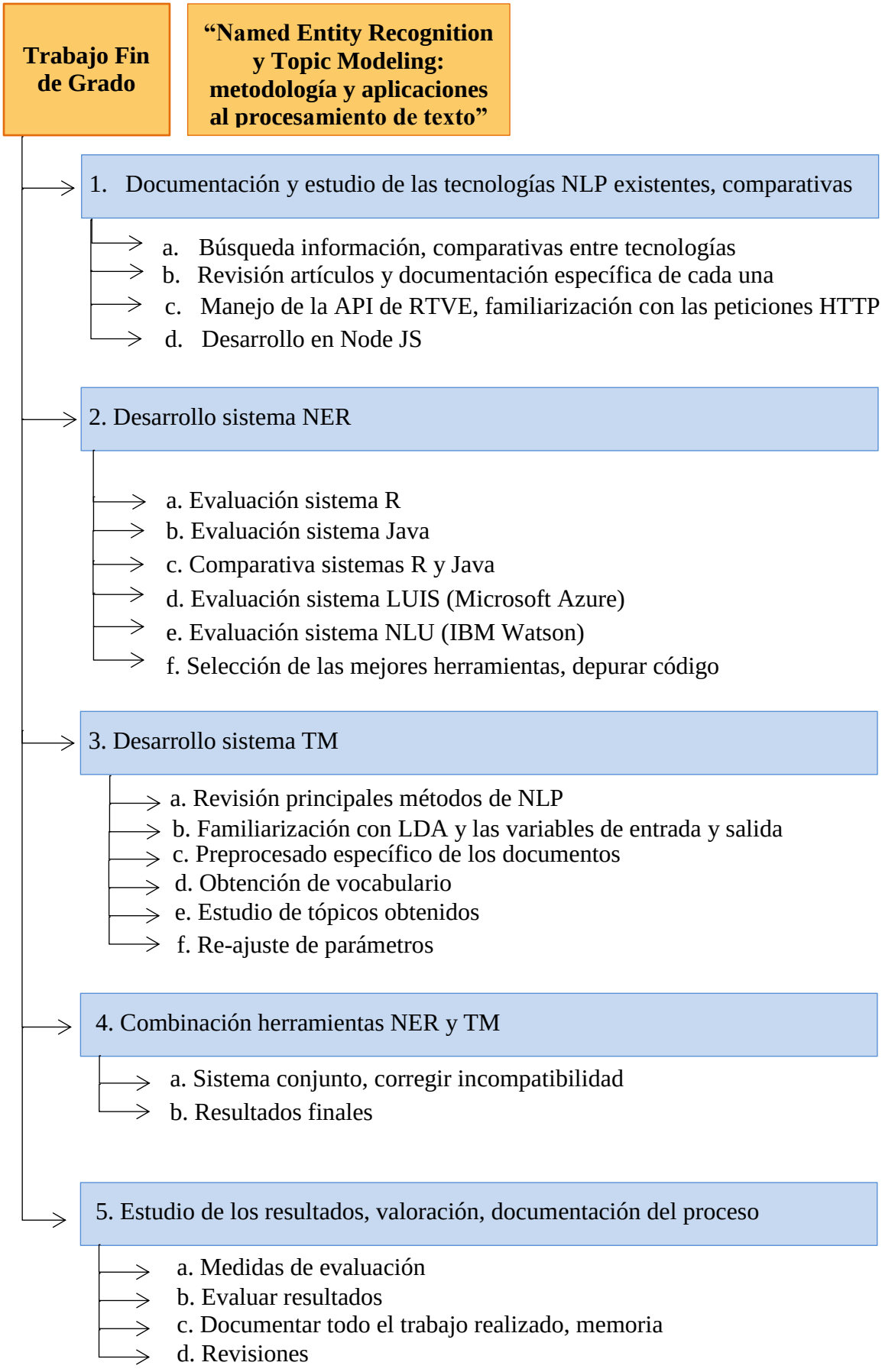
5.1. Planificación trabajo:

Este proyecto consta de cinco grandes bloques:

1. Documentación y estudio de las tecnologías *NLP* existentes, comparativas
2. Desarrollo sistema *NER*
3. Desarrollo sistema *TM*
4. Combinación de las herramientas *NER* y *TM*
5. Estudio de los resultados, valoración, documentación del proceso

A continuación, se desglosan las subtareas de los cinco bloques, que servirán de hitos para seguir el progreso del trabajo (Fig. 5.1).

Fig. 5.1: Diagrama de bloques y tareas del trabajo.



5.2. Diagrama de Gantt

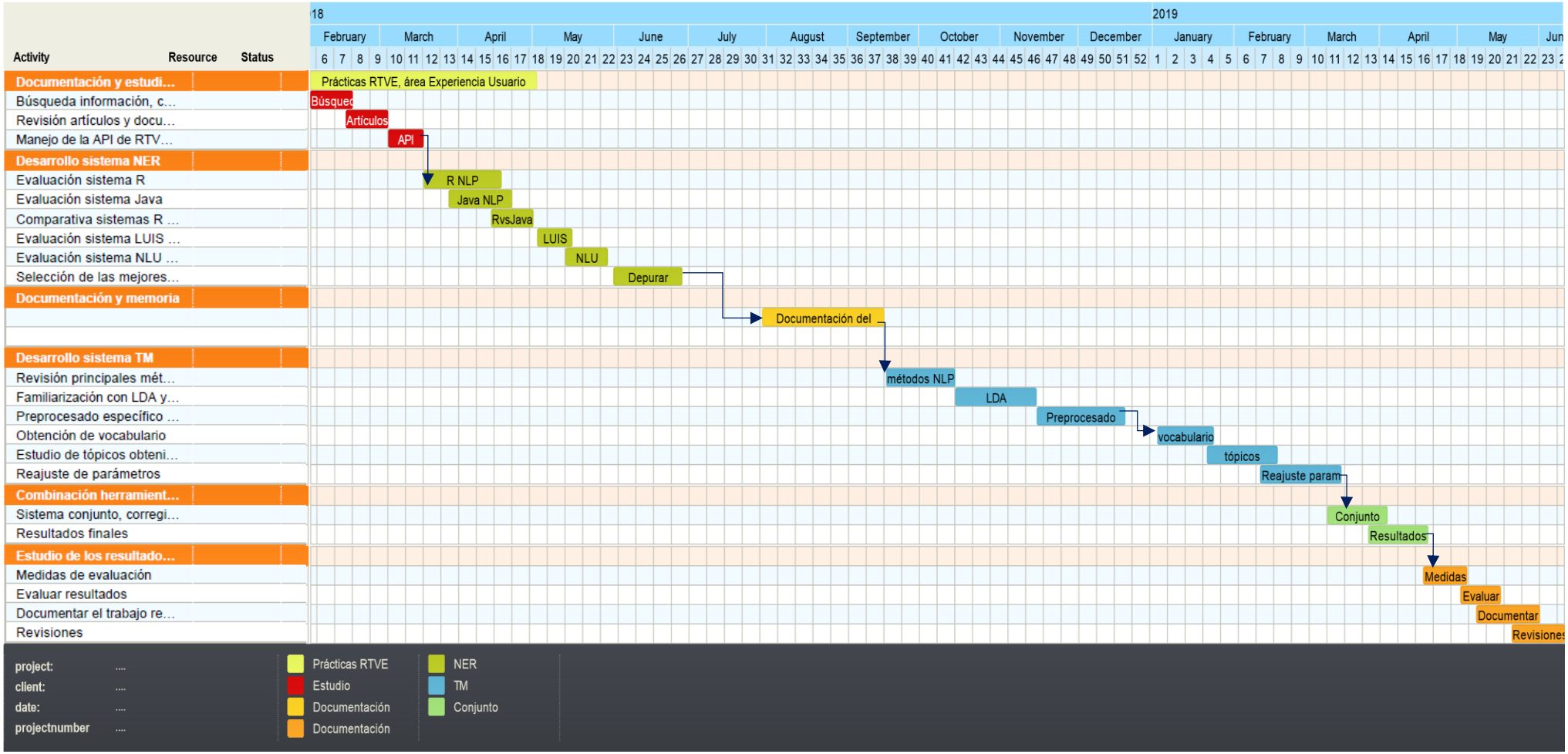


Fig. 5.2: Diagrama de Gantt con el desarrollo en el tiempo de las etapas del proyecto. La lista de actividades (panel izquierdo de la imagen) corresponde con las tareas agrupadas en bloques (Fig. 5.1), y en la retícula derecha, su desarrollo temporal. Realizado en: https://www.tomsplanner.com/

5.3. Duración del proyecto:

El proyecto se ha desarrollado entre los meses de febrero de 2018 y junio de 2019.

Tabla 5.1: Cuadrante de horas dedicadas al proyecto.

	Estudiante	**Tutor**	**RTVE**
Febrero a mayo 2018, 16.8 semanas, 84 días	5h/día, 5d/sem, = 420 h	2 reuniones/mes, 45 min = 9.5 h	1 reunión/sem, 16.8 sem = 25 h
Junio a agosto 2018, 13 semanas	7h/día, 2d/sem = 182 h	1 reunión 3h = 3h	-
Septiembre a diciembre, 17 semanas	3h/día, 1d/sem = 51 h	2 reuniones/mes 1h = 8h	-
Enero 2019 5 semanas	4h/día, 1d/sem = 20 h	1 reunión/mes = 1h	-
Febrero a abril 2019 62 días	6h/día, 3d/sem = 186 h	3 reuniones/mes = 3h	-
Mayo, junio 2019 42 días	9h/día, 6d/sem = 378 h	1.5 reun/sem = 18 h	-
Total horas:	1250 h	200 h	25 h

5.4. Presupuesto

Este proyecto ha consistido en el estudio y validación de algunas herramientas *NLP* mediante la implementación de algunas tecnologías. Para ello, los principales activos han sido los equipos en que se han ejecutado y desarrollado los códigos y las personas que han trabajado en ello.

En primer lugar, se contemplan los costes invariables ante cambios en los niveles de producción (**costes fijos**), en los que la inversión que hay que hacer es la misma independientemente de cuánto vaya a durar el trabajo o el volumen de producción:

Equipos.

Este proyecto se ha desarrollado empleando dos **ordenadores** para el desarrollo del trabajo, y otros dos para la validación del mismo, aunque hubiera podido bastar con uno o un par de equipos.

Considerando un coste inicial de unos 600€ para el ordenador, y una vida útil de 8 años, la devaluación que sufre es de 75€/año. Este proyecto se ha desarrollado durante un periodo de 15 meses, lo que supone una devaluación **93,75€** por la utilización y ejecución de sistemas, más el coste de las posibles reparaciones que hicieran falta. (En el anecdótico

caso del desarrollo real de este proyecto, por un fallo fatal en la tarjeta de vídeo del primer ordenador de desarrollo, hizo falta la adquisición del segundo equipo de desarrollo, lo que duplicaría estos costes y sumaría el precio de la compra del nuevo equipo).

Para el o los ordenadores de evaluación (tutor de las prácticas de la universidad y director del proyecto en RTVE), los costes podrían ser equiparables, aunque algo superiores (se consideran unos ordenadores con prestaciones ligeramente superiores, estimables en entre un 20% y un 50% del precio).

Asimismo, se ha empleado una **impresora** cuyo coste ronda los 200€, y que puede tener una vida útil de unos 4 años. A ese ritmo, la devaluación es de 50€ al año, y en un periodo de año y medio, **75€**.

Sistema operativo.

Sin entrar en el detalle de los programas, el propio **sistema operativo** del ordenador tiene un coste. En el caso de haber adquirido un equipo con el sistema operativo incorporado, este precio no ha de considerarse como un plus, pero si es un ordenador vacío o si se ha de resetear uno antiguo, debe obtenerse una licencia.

Tabla 5.2: Comparativa Sistemas Operativos disponibles y precios orientativos.

S.O.	Plan / paquete	Precio
Windows	Windows Home	145,00€
	Windows Pro	259,00€
	Windows Pro for Workstations	439,00€
Mac	Mac OS	979,00€ en equipo Windows, 1.399,00€ en equipo Mac
Linux	Ubuntu, Mint, Arch, Deepin, Fedora, Debian, openSUSE	Gratis, abierto

Licencias software.

Las principales tecnologías que se han empleado han sido de código abierto y licencia libre (open source, gratis). Estas han sido:

- Soporte Java, JDK.
- Node JS, entorno Visual Studio Code.
- R, con entorno RStudio.
- Python, entorno Anaconda, con Jupyter Notebook.
- Terminal de comandos de Windows, cmd.

- Librerías Python: NLTK, Gensim, Spacy.
- Librerías R: NLP, openNLP, RJava, RWeka, Magrittr, Qdap, openNLPmodels
- Librerías Java: Apache OpenNLP

Del mismo modo, también se procedió a evaluar algunas tecnologías propietario:

- Azure Microsoft -> Language Understanding Intelligence Service (LUIS)
- Watson IBM -> Natural Language Understanding (NLU)

En ambas herramientas los planes son seleccionables, en función de la potencia requerida y funcionalidades que se deseen emplear, se puede contratar un paquete software u otro.

Por otro lado, están los costes que dependen de la magnitud de la producción, los denominados **costes variables**:

Consumo eléctrico.

Para el trabajo con el ordenador, se han requerido recursos eléctricos para alimentar el equipo, la iluminación del puesto de trabajo, y la impresora con la que se imprimía la documentación y memoria.

- La *CPU* del ordenador consume diariamente 50,56 W x 5h / 1000 = 0,2528 kWh. Considerando que está encendido durante toda la jornada de trabajo (~800h). Esto suponen 202,24 kW.

- La lámpara es un flexo halógeno, que consume 48 W. Se enciende sólo hasta las 11h de la mañana, o desde las 16h de la tarde, por lo que se considera la mitad de la jornada (~400h). Esto suponen 19,2 kW.

- La impresora es de 2011, consume 10 W. Se considera encendida toda la jornada (~800h), con lo que se llegan a 8 kW.

Según las comparativas entre la compañía Iberdrola, Endesa, Gas Natural Fenosa, Edp, Podo y Repsol [37], el coste actual del kWh en España oscila entre los 0,1170€/kWh y los 0.1449 €/kWh, por lo que se toma un precio intermedio de 0.134 €/kWh, que se ha mantenido estable en los últimos dos años.

Esto implica un coste total por electricidad de (202,24 kW + 19,2 kW + 8 kW) x 0,134 = 229,440 kW x 0,134 = **30,74 €**.

Impresiones.

La impresora se ha empleado para imprimir algunos fragmentos de código para su evaluación, así como ciertas comparativas, manuales, o capítulos de la memoria.

Un cartucho de tinta permite imprimir unas 180 páginas, por un precio de ~13€ el cartucho. Igualmente, un paquete de 500 hojas cuesta ~3€. En total, no se han superado las 200 páginas impresas ni han sido imágenes con mucho gasto de tinta, por lo que se considera un único cartucho y un paquete de hojas, en total, **16€**.

Se añade también un bolígrafo porque así se pueden hacer anotaciones en las páginas impresas (0,30€).

Recursos humanos.

Para este proyecto se han requerido tres papeles fundamentales:

- Estudiante: último año de Ingeniería de Sistemas Audiovisuales
- Tutor: profesor ingeniero de la Universidad Carlos III, doctorando
- Co-tutor (director del Área de Experiencia del Usuario de RTVE)

Se considera una retribución por horas en función de la cualificación de cada puesto, y se desglosa la carga de trabajo de cada uno:

Tabla 5.3: Presupuesto para recursos humanos.

Puesto	Horas de trabajo	Retribución	Total
Estudiante	1.237 h	12	14.844 €
Tutor doctorando	200 h	23	4.600 €
Director área RTVE	25 h	30	750 €
		Total	**20.194 €**

Presupuesto total.

Sumando todos estos costes, el presupuesto requerido para este proyecto es el siguiente:

Tabla 5.4: Desglose del presupuesto total.

CONCEPTO	COSTE
Costes fijos materiales	
Devaluación PC	93,75 € x2 unidades
Devaluación impresora	75 €
Sistema Operativo	Supuesto nulo
Licencias software	Supuesto nulo
Costes variables	
Electricidad	30, 74 €
Impresiones	16, 00 €
Recursos humanos	20.194, 00 €
SUBTOTAL	20.503, 54 €
IVA (21%)	4.305, 74 €
TOTAL	24.809, 28 €

6. CONCLUSIONES Y LÍNEAS FUTURAS

En este proyecto, se han evaluado distintas herramientas de procesamiento de lenguaje natural, realizando un estudio comparativo de las tecnologías existentes en el mercado, e implementando un sistema de reconocimiento de entidades nombradas (*NER*) en los lenguajes de R y Java, y un sistema de modelado de tópicos (*TM*) en lenguaje Python.

6.1. Conclusiones

Durante este proyecto, se ha llevado a cabo un estudio del paradigma actual del procesamiento del lenguaje natural. En primer lugar, se ha realizado un estudio de los sistemas existentes y las tecnologías actuales que se están desarrollando e investigando, como parte del estado del arte. Se ha realizado una revisión histórica de las primeras tecnologías y su desarrollo hasta el momento presente, presentando las tecnologías actuales y las áreas de investigación (Tablas 2.1 y siguientes).

A continuación, el proyecto se ha dividido en dos ramas de *NLP*, que luego convergerán. En la primera de ellas se exploró el Reconocimiento de Entidades Nombradas, a través de la implementación de una persona, entidades y entidades extractoras de ubicaciones. Se han empleado para ello los documentos de subtitulado de RTVE. Se han elegido cuatro sistemas para comparar los beneficios de cada uno: dos de ellos basados en bibliotecas y lenguajes de código abierto, y dos de ellos, software propietario. Se completó el desarrollo en tecnologías R y Java a través de la biblioteca OpenNLP de Apache (Tabla 3.1 para la lista completa de herramientas utilizadas), con resultados satisfactorios (presentados en la sección 4: Tabla 4.3 y siguientes), aunque las otras dos tecnologías evaluadas, LUIS de Azure Microsoft y NLU de IBM Watson, no se implementaron completamente. La etapa de desarrollo que se alcanzó en cada uno de ellos se discute en la sección 3.4.

La segunda rama fue *Topic Modeling* para el mismo conjunto de documentos. Después de una breve revisión de los mecanismos existentes se eligió el modelo *LDA*. Desde el corpus de documentos hasta el modelo *LDA*, se utilizan las bibliotecas NLTK y Gensim de Python (tabla completa de herramientas utilizadas en la Tabla 3.1). Este modelo crea un diccionario de palabras significativas que aparecen en el corpus y hace el recuento de las apariciones. Usando esta información, se obtienen 15 temáticas, caracterizadas por una lista de términos significativos, y se exploran distintas visualizaciones en *pyLDAvis*. La calidad de estos modelos no es cuantificable, pero a través de diferentes medidas exploradas (sección 4.2.2) los resultados se encuentran satisfactorios.

El propósito de estos dos sistemas es complementarse entre sí para la extracción de información más completa de los datos textuales. Los resultados de ambos sistemas se contrastan (sección 4.2.3), y cumplen los objetivos deseados.

Hay varias líneas abiertas para el trabajo futuro, que se analizarán en el capítulo 6.2. Estas mejoras perfeccionarían el funcionamiento del sistema desarrollado, que se utilizaría en las plataformas de contenido audiovisual de RTVE (sistema de televisión bajo demanda "Televisión a la Carta", aplicación móvil y televisión inteligente). Sería útil para identificar el contenido en los vídeos (subtitulados) y documentos, para que se etiqueten automáticamente en categorías y se clasifiquen para mejorar la recuperación de archivos

de cierto tema o entidad. Además, esto mejorará la experiencia del usuario y permitirá nuevas recomendaciones a partir del contenido al que se accedió anteriormente.

6.2. Líneas futuras

Dentro de las líneas abiertas de continuación de este proyecto se proporcionan algunas ideas que podrían explorarse:

- Incorporación de otras tecnologías NLP.

Este proyecto se ha basado en el estudio e implementación de dos tecnologías de procesamiento del lenguaje natural, que son el reconocimiento de entidades nombradas en un texto (*NER*), y el modelado de *topics* o temáticas (*TM*). Estos dos sistemas podrían complementarse con implementaciones de otras tareas de *NLP*, como por ejemplo el análisis de los sentimientos del autor de un texto (*Sentiment Analysis*) o la incorporación de un módulo *OCR* (reconocimiento óptico de caracteres), de forma que una imagen de un texto pudiera convertirse a texto plano antes de comenzar a trabajar con él. Otras funcionalidades que podrían implementarse son traducción del texto a voz (*Text to Speech*), o viceversa (*Speech To Text*). Todas estas funcionalidades y tareas del *NLP* se recogen en la Tabla 2.1 de este trabajo.

- Mejoras en el sistema de reconocimiento de entidades.

Junto con la investigación de las distintas tecnologías *NER* existentes en el mercado (Tabla 2.2), en este proyecto se seleccionaron 4 de ellas para evaluarse mediante una implementación práctica (Tabla 3.1). Por la envergadura de esta sección de análisis no se pudieron completar las implementaciones de todas ellas. Otra línea abierta sería continuar el desarrollo en las otras tecnologías seleccionadas o incluso evaluar algunas de las otras existentes (Tabla 2.2).

Otra línea interesante de desarrollo es el análisis de entidades en imagen o vídeo, que combinaría el reconocimiento y manejo de textos con el reconocimiento de objetos en imagen o vídeo. Esto supondría una incursión en el ámbito de *Computer Vision* (visión por ordenador), y tecnologías de visión artificial que combinaría el reconocimiento de información en objetos o de rostros con el de información escrita.

- Mejoras en el sistema de modelado de *topics*.

En el modelado de *topics* se ha optado por una implementación única en Python a través de las librerías y funcionalidades detalladas en la Tabla 3.1. Existe la posibilidad de explorar otras tecnologías existentes, por ejemplo, basadas en alguno de los métodos históricos de TM (apartado 2.2.3), como los *Correlated Topic Models*.

Por supuesto, también podrían estudiarse o desarrollarse otras métricas de error para la evaluación de la calidad de los modelos de *topics*, ya que, como se ha explicado previamente (apartados 4.1.2 y 4.2.2), no existe un procedimiento estandarizado para su cuantificación. También podrían perfeccionarse los sistemas de medición de la coherencia para el corpus de documentos empleado y en idioma español, ya que en la actualidad estas

medidas están optimizadas para corpus externos muy distintos al empleado y en lengua inglesa.

- Volumen de datos.

Los sistemas que se han desarrollado han sido implementados sobre un corpus de documentos de subtitulado procedentes de la API de RTVE, obtenidos de 10 temáticas. Una mejora que permitiría tener un sistema más potente sería la reescala del sistema para entrenar los modelos de reconocimiento de entidades y obtención de temáticas a partir de una base de datos más grande de documentos. Igualmente, la implementación de este sistema sobre una plataforma web o sobre la API permitiría evaluar el funcionamiento en condiciones más amplias.

- Validación cruzada de parámetros empleados.

Dentro de los ámbitos que iban a manejarse en este proyecto se valoró incorporar una validación de los parámetros empleados en *NER* y *TM* a través de *cross-validation* (validación cruzada). Para *TM* se buscaría validar los parámetros para obtener una mejor coherencia. Por razones de recursos para este proyecto limitados en el tiempo finalmente no llegó a incorporarse esta implementación en el sistema final, por lo que se propone como línea abierta para el perfeccionamiento de los módulos existentes.

- Sistema conjunto

Complementariamente a la combinación de los resultados de *NER* y *TM* se pueden explorar otros sistemas para relacionar las entidades y los tópicos aprendidos en un mismo corpus. Un ejemplo podría consistir en añadir las entidades reconocidas al modelo generativo del modelo de tópicos.

Otras mejoras podrían consistir en explorar modelos de *topics* más complejos, que permitan aprender sobre las relaciones entre los términos característicos y así, en consecuencia, entre las entidades identificadas.

Además, se podría experimentar con otras métricas de similitud entre entidades, por ejemplo, empleando *Word Embeddings*.

- Posteriores desarrollos para el contexto concreto de Televisión a la Carta de RTVE.

Este proyecto surgió como una primera aproximación a una tecnología de *Text Mining* para la mejora de la experiencia de usuario dentro del proveedor de contenidos audiovisuales RTVE, y en concreto, dentro de la web y aplicación para móvil y Smart TV.

Si este trabajo se hubiese planteado para un desarrollo temporal superior, se habría investigado la forma de integrar los sistemas desarrollados dentro de las plataformas existentes de RTVE con experiencia de usuario (web, app y Smart TV). Este proyecto no se ha desarrollado hasta este punto, pero a través de un proceso de validación de los códigos y evaluación de los sistemas, podría llegar a implementarse próximamente o corregirse lo que fuese necesario para ello.

7. REFERENCIAS

1. Motivación

[1] P. LYMAN y H. R. VARIAN. *How much information?* University of California, Berkeley: UC Press, 2000.

[2] D. REINSEL et al. "The expanding digital universe: a forecast of worldwide information growth through 2010". White paper, IDC. Mar. 2007. http://core.xsomo.com.jm/images/web/File/white%20papaers/Expanding_Digital_Universe_IDC_WhitePaper_022507.pdf

[3] F. GANTZ y D. REINSEL. "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east". *IDC iView: IDC Analyze the future*, vol. 2007, no 2012, pp. 1-16, dic. 2012. https://www.emc-technology.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf

[4] W. VAN DER AALST. "Data science in action", en *Process Mining*. Springer, Berlin: Heidelberg, abr. 2016, pp. 3-23.

[5] TURING, Alan M. Computing machinery and intelligence (1950). The Essential Turing: The Ideas that Gave Birth to the Computer Age. Ed. B. Jack Copeland. Oxford: Oxford UP, 2004, p. 433-64.

2. Planteamiento del problema

2.1. Tecnologías NLP

[6] D. M. Blei, Figura: Fundamentos modelos probabilísticos de TM "Probabilistic Topic Models", *Communications of the ACM*, vol. 55, nº4, pp. 77-84. Princeton University, Abr, 2012.

[7] B. Ravi, "Entity Extraction—Demystifying Rasa NLU", *Hackernoon*. Aug. 2008 https://hackernoon.com/entity-extraction-demistifying-rasanlu-part-3-13a460451573 (acceso: mayo 2019)

2.1.1. NER

[8] "Top 27 free software for Text Analysis, Text Mining, Text Analytics". *Predictive Analytics Today*. https://www.predictiveanalyticstoday.com/top-free-software-for-text-analysis-text-mining-text-analytics/ (acceso: mayo 2019).

[9] "Natural Language Processing". *Wikipedia*. https://en.wikipedia.org/wiki/Natural_language_processing (acceso: abril 2019)

[10] Tabla 7.1: Documentación oficial tecnologías NER referidas: (acceso: mayo 2019).

Tecnologías	Documentación
GATE	https://gate.ac.uk/#
Apache OpenNLP	https://opennlp.apache.org/docs/1.9.1/manual/opennlp.html
Stanford NER	https://nlp.stanford.edu/software/CRF-NER.html
NLTK	https://www.nltk.org/
Snowball	http://snowball.tartarus.org/
Spacy	https://spacy.io/usage/spacy-101#features
Weka	https://www.cs.waikato.ac.nz/ml/index.html
R: RWeka, TM	https://cran.r-project.org/web/views/NaturalLanguageProcessing.html
TextBlob	https://textblob.readthedocs.io/en/dev/
Carrot2	https://doc.carrot2.org/
Gensim	https://radimrehurek.com/gensim/
NERD	http://nerd.eurecom.fr/documentation
Google Cloud NLP + API	https://cloud.google.com/natural-language/docs/
Watson NLU	https://cloud.ibm.com/docs/services/natural-language-understanding?topic=natural-language-understanding-getting-started#getting-started
Azure Language Understanding	https://azure.microsoft.com/es-es/services/cognitive-services/language-understanding-intelligent-service/

[11] P. Joshi. “An NLP approach to Mining Online Reviews using Topic Modeling (with Python codes)”, *Analytics Vidhya*, oct. 2018. https://www.analyticsvidhya.com/blog/2018/10/mining-online-reviews-topic-modeling-lda/ (acceso: abril 2019)

2.1.2. TM

[12] S. Deerwester et al. “Indexing by Latent Semantic Analysis”. *Journal of the American society for information science*, vol. 41, no 6, pp. 391-407. 1990.

[13] Figura Matriz de ocurrencias para aplicar LSA, “Latent Semantic Analysis”, Wikipedia: https://en.wikipedia.org/wiki/Latent_semantic_analysis (acceso: abril 2019)

[14] S. C. Deerwester, et. al. *Computer information retrieval using Latent Semantic structure.* U.S. Patent No 4,839,853, 13 Jun. 1989.

[15] A. Navlani, Figura Descomposición de la matriz en valores singulares, “Latent Semantic Analysis using Python” en *Datacamp Tutorials*, oct. 2018. https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python (acceso: abril 2019).

[16] D. M. Blei, A. Y. Ng y M. I. Jordan. “Latent dirichlet allocation”, en *Journal of machine Learning research*, vol. 3, pp. 993-1022. Ene. 2003.

[17] T. L. Griffiths et al. “Hierarchical topic models and the nested Chinese restaurant process”. En *Advances in neural information processing systems*. pp. 17-24, 2004.

[18] D. M. Blei, et al. “A correlated topic model of Science”. *The Annals of Applied Statistics*, vol. 1, no 1, pp. 17-35. 2007.

[19] J. D. Lafferty y D. M. Blei. Figura del grafo de *topics* para artículos de la revista *Science*. “Correlated topic models”. En *Advances in neural information processing systems*. p. 147-154. 2006.

[20] D. M. Blei y J. D. Lafferty. “Dynamic topic models”. En *Proceedings of the 23rd international conference on Machine learning*. ACM, pp. 113-120. 2006.

[21] D. M. Blei y J. D. Lafferty. Figura para ejemplo de *TM* dinámicos. “Dynamic topic models.” ICML 2006.

[22] M. Rathore, “Dynamic Topic Models Tutorial”, en *GitHub*. https://markroxor.github.io/gensim/static/notebooks/ldaseqmodel.html

[23] A. Srivastava y C. Sutton. *Autoencoding variational inference for topic model*. arXiv preprint arXiv:1703.01488, Mar. 2017.

2.2. Marco legislador

[24] Código de Derecho Audiovisual. BOE. https://www.boe.es/legislacion/codigos/codigo.php?id=168_Codigo_de_Derecho_Audiovisual&modo=1 (acceso: mayo 2019).

[25] Ley 7/2010, de 31 de marzo, General de la Comunicación Audiovisual. «BOE» núm. 79, de 1 de abril de 2010. Referencia: BOE-A-2010-5292

[26] *Control de Contenidos*, Comisión Nacional del Mercado y la Competencia, https://www.cnmc.es/ambitos-de-actuacion/audiovisual/control-de-contenidos (acceso: mayo 2019)

[27] Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal (LOPD), https://www.boe.es/buscar/doc.php?id=BOE-A-1999-23750 (acceso: mayo 2019)

[28] Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales (LOPD-GDD) https://boe.es/boe/dias/2018/12/06/pdfs/BOE-A-2018-16673.pdf (acceso: mayo 2019)

[29] Ley 2/2019, de 1 de marzo (ordenamiento jurídico español la Directiva 2014/26/UE del Parlamento Europeo y del Consejo, de 26 de febrero de 2014) http://noticias.juridicas.com/base_datos/Privado/639062-l-2-2019-de-1-mar-modifica-el-texto-refundido-de-la-ley-de-propiedad-intelectual.html (acceso: mayo 2019)

2.3. Marco socioeconómico

[30] S. Bertoni, "Exclusive Interview: How Jared Kushner Won Trump The White House", *revista Forbes*. Dic. 2016. https://www.forbes.com/sites/stevenbertoni/2016/11/22/exclusive-interview-how-jared-kushner-won-trump-the-white-house/#3b5e0d763af6 (acceso: mayo 2019)

[31] M. E. Porter. *Análisis Porter de las cinco fuerzas*. Jul.-ago. 1979.

[32] V. Khandpur. Figura: Ciclo de vida de una tecnología. "How Technology Life Cycle Helps In Patent Portfolio Maintenance?" Greyb Research. https://www.greyb.com/technology-life-cycle-helps-patent-portfolio-maintenance/ (acceso: mayo 2019)

[33] C. Y. Wong, V. G. R. Chandran y B. K. Ng. Figura: Curva S de difusión de una tecnología. "Technology Diffusion in the Telecommunications Services Industry of Malaysia", en *Information Technology for Development.* Malasia. Sep. 2014, https://www.researchgate.net/figure/The-S-curve-of-diffusion-of-technology_fig2_271668429 (acceso: mayo 2019)

3. Diseño solución

4. Resultados

[34] J. Chang et al. "Reading tea leaves: How humans interpret topic models". En *Advances in neural information processing systems*. pp. 288-296. 2009.

[35] M. Röder, A. Both y A. Hinneburg. "Exploring the space of topic coherence measures". En *Proceedings of the eighth ACM international conference on Web search and data mining*. ACM, pp. 399-408. 2015.

[36] Figura: producto escalar de dos vectores. *Producto escalar*, Wikipedia, https://es.wikipedia.org/wiki/Producto_escalar (acceso: mayo 2019)

5. Organización

5.4. Presupuesto

[37] Comparativa tarifa luz-hora, Selectra. https://tarifaluzhora.es/info/precio-kwh (consulta: abril 2019)

Anexo: Glosario

[38] Figuras sobreajuste, *Wikipedia*, https://es.wikipedia.org/wiki/Sobreajuste (acceso: mayo 2019)

[39] D. M. Blei, A. Y. Ng y M. I. Jordan, "Latent Dirichlet Allocation", Journal of Machine Learning Research, vol. 3, n.º 3, pp. 993-1022, ene. 2003. http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

[40] C. H. Papadimitriou, et al. "Latent semantic indexing: A probabilistic analysis." *Journal of Computer and System Sciences*, vol. 61, no 2, pp. 217-235, oct. 2000.

[41] S. Deerwester, S. T. Dumais, George W. Furnas, Thomas K. Landauer y Richard Harshman. "Indexing by latent semantic analysis", *Journal of the Association for Information Science and Technology*, vol. 41, n.º 6, pp. 391-407, sept. 2003.

ANEXO A. GLOSARIO DE TECNICISMOS

Big Data [Apartado 1.1]

Macrodatos o datos masivos. Conjuntos de datos con alta **v**elocidad de crecimiento, **v**ariabilidad en la información (complejidad) y gran **v**olumen. Por el gran crecimiento de información registrada a cada instante en internet y la globalización de las comunicaciones, se hacen necesarios nuevos enfoques para gestionar estas grandes cantidades de información de forma potente y eficiente, frente a las bases de datos tradicionales y los algoritmos basados en conjuntos de reglas.

Information Retrieval [Apartado 2.2.1]

Proceso de organización de la información (comúnmente, textos) y desarrollo de algoritmos que permitan hacer solicitudes para recuperar los datos de interés. En estos sistemas se fundamentan por ejemplo los buscadores de internet, que, a partir de una consulta, recuperan resultados relacionados con las palabras o condiciones introducidas.

Text Mining

Procesamiento para derivar información relevante ("*high quality information*") a partir de texto plano, sin etiquetas u otras identificaciones. Busca localizar conceptos, patrones, temáticas, palabras clave, estructuras, y otros atributos del texto. La información *"high quality*" se considera tal en atención a su relevancia, novedad e interés.

Es un ámbito de la minería de datos ("***Data Mining***"), que trata de extraer información subyacente en estadísticas, imágenes, textos, etc. Muchas veces, la información que se trata de extraer es información que las personas abstraemos sin darnos cuenta o que relacionamos involuntariamente, mediante procedimientos cognitivos de los que no somos plenamente conscientes.

Conceptos relacionados: "*Text Analysis*".

***Machine Learning*, aprendizaje automático**

Existen tres tipos de aprendizaje para la solución de problemas en una máquina:

- Modelos basados en reglas (aproximación convencional, instrucciones)
- Modelos basados en etiquetas (aprendizaje supervisado, en que durante el aprendizaje el modelo se reajusta comparando los resultados que obtiene con las soluciones que debería obtener)
- Modelos no supervisados.

Un algoritmo de *Machine Learning* o aprendizaje automático busca un patrón de "solución deseada" a partir de un gran número de ejemplos, sin que el programador decida en qué características de los datos debe fijarse. Las técnicas de análisis no supervisado de *Machine Learning* se emplean para resolución de problemas de *Data Mining*.

Sobreajuste, subajuste [Apartado 3.4]

Dentro del entrenamiento de un sistema de *Machine Learning*, existen unos efectos por un tratamiento incorrecto de los datos de entrenamiento. A continuación, se explican los efectos del sobreajuste a los datos de entrenamiento (*overfitting*), y del subajuste (*underfitting*).

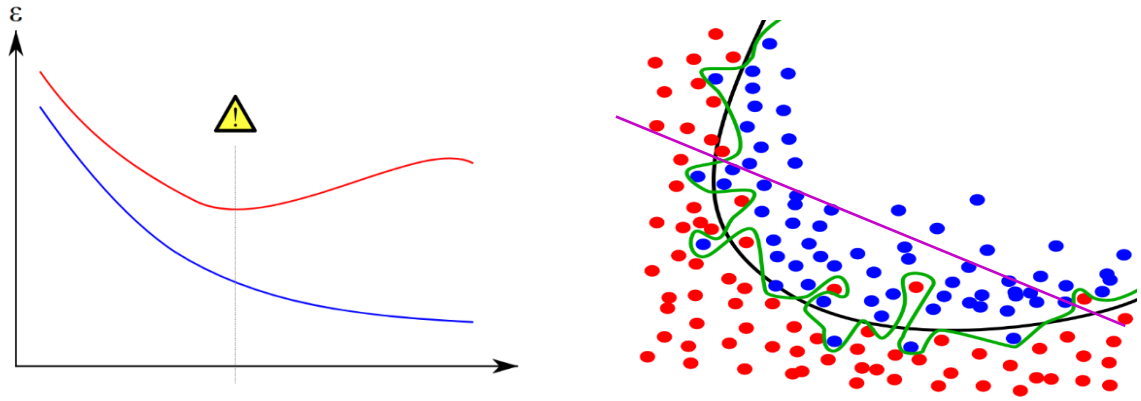


Fig. A.1: Efectos de un sobreajuste: (Izda:) Gráfica que muestra el descenso de la tasa de error sobre los datos de entrenamiento (azul) y para nuevos datos, conjunto de test (rojo). (Dcha:) Resolución de un problema de decisión binario con solución subajustada (morado), adecuada (negro) y sobreajustada (verde). Fuente: Wikipedia [38]

En la gráfica de la derecha, se puede apreciar cómo desciende el error conforme se repite más veces el entrenamiento con esas muestras, hasta hacerse casi nulo. Este comportamiento correspondería a la curva verde de solución de la izquierda, en que el sistema se empeña en reducir al máximo el error durante el entrenamiento, y se denomina **sobreajuste** (overfitting). Esto no es un comportamiento deseable porque significa que para nuevas muestras que estén cerca de las de entrenamiento, pero no exactamente en el mismo sitio, probablemente el sistema las clasificará mal. El sistema se ha acostumbrado a las particularidades de las muestras de entrenamiento y ha perdido capacidad de generalizar. Es por esto que, pasado el umbral de la gráfica de la izquierda, el error de test (nuevas muestras) de la curva roja vuelve a crecer.

En el extremo contrario tenemos la zona cercana al eje vertical de la gráfica izquierda, y la curva morada de división de las clases. Por haber entrenado demasiado poco, la clasificación que hace el sistema es muy básica, y se pierden muchas muestras, hay un error alto tanto para entrenamiento como para validación. Esto se denomina **subajuste** (*underfitting*).

***Natural Language Processing* (*NLP*):** [Apartado 2.2]

"Procesamiento de lenguaje natural". Conjunto de técnicas para obtener de forma automática en un ordenador la información implícita en un texto que los seres humanos inferimos sin darnos cuenta. Esta información puede ser el tema del texto, las entidades que aparecen (lugares, organizaciones, personas), la estructura de la información (párrafos, capítulos, notas al pie), o incluso el estado de ánimo que se trasluce en una frase.

Entre los lenguajes más empleados para su implementación están Python, con su Natural Language ToolKit, y el lenguaje de computación estadística y gráfica R. Este último

permite interactuar con otros lenguajes como C, C++ y Java, por lo que es posible en un código R incluir librerías en esos lenguajes de nivel inferior que son más rápidos, conservando las ventajas de la programación funcional de R y sus muchas otras librerías para análisis de datos.

***Named-Entity Recognition* (*NER*):** [Apartado 2.2.2]
"Reconocimiento de entidades nombradas". Conjunto de algoritmos y métodos de identificación de entidades, con el fin de poder clasificarlas en categorías (p.ej. nombres de personas, organizaciones, lugares, fechas) y así ser más fácilmente accesibles.

Sinónimos: "identificación de entidades". En inglés: "entity identification", "entity chunking", "entity extraction".

***Topic Modeling* (*TM*):** [Apartado 2.2.3]
Modelo probabilístico generativo para extraer funciones de distribución que caractericen las apariciones conjuntas (coocurrencias) de ciertas palabras (*tokens*) en documentos de la misma temática (*topic*). Es una técnica concreta de *Text Mining*.

Mediante un modelo estadístico se localizan los términos (en inglés, "*terms*") para deducir el tema o ámbito ("*topic*") al que se está refiriendo un texto. Se emplea la identificación de patrones o estructuras en él, de palabras clave, y de coincidencias de términos que aparecen juntos en textos que tratan de un mismo ámbito.

Dos de las técnicas de *Topic Modeling*:

- *Latent Dirichlet Allocation* (LDA), en que cada documento pertenece a una temática (*topic*) con una probabilidad. Cada *topic* agrupa una serie de palabras clave relacionadas semánticamente. De cada documento se obtienen los porcentajes de coincidencia con las palabras clave de un *topic*. [##]

- Latent Semantic Indexing (LSI), que descompone los documentos en valores singulares (SVD) para aprender tópicos.

***Perplexity*:** [Apartados 4.1.2, 4.2.2]
Parámetro empleado como medida de calidad en TM. Resultado al que converge el modelo. Estimación de la verosimilitud (no es una tasa de acierto porque no puede calcularse dicho valor en TM). Sirve para encontrar cuándo se estabiliza el modelo, viendo la evolución temporal.

Se basa en la verosimilitud, que se aproxima con la cota por no poder calcularse el valor exacto.

ANEXO B. SUMMARY

In a world increasingly globalized, the amount of information that we generate grows exponentially. In the present moment, it is estimated in 2.5 quintillion of bytes daily generated, that would be equivalent to the number of neurons that would add 250 million human brains. Internet of Things and social media multiply the amount of information accessible and generated by each individual user, requiring increasingly powerful processing and storage systems. To handle these massive amounts of information (Big Data) that are characterized by their speed (velocity), variety and volume, conventional systems are not enough.

As a new proposal to solve old and new problems, Artificial Intelligence develops Machine Learning. It consists of the development of systems able to identify unknown patterns from input data, to build a mathematical model that structures the solution for future problems to solve.

Within this great area, there are specific fields differentiated according to the problems on which they focus.

Natural Language is a kind of language originated spontaneously among human beings, for the purpose of communicating. This kind of written or spoken speech differs from formal languages (i.e. formal logic, mathematic logic) or programming languages. For the automatic analysis of natural languages, computer techniques are applied, known as Natural Language Processing.

The usefulness of these systems lies in the streamlining of the processing of spoken texts or discourses, the ability to extract some implicit information in human language, of which many times we are not aware: distinction of meanings of the same word depending on the context, topic that is being discussed, references to information previously known by both interlocutors, word games, mood of the person speaking, opinion on a topic, etc. All this information implicit in the discourse differentiates a strict textual understanding of what has been said exactly from a deeper understanding of what is wanted to be said.

This project's motivation is to carry out on some existing tools within this field, and to develop a system capable of extracting the most relevant entities and the topic or topics that are being discussed in a text. The documents will then be characterized by the keywords that appear in them and by the common scope they all have, according to probabilistic models. In this way, the system will be able to return documents related to a theme or entity.

B.1 Milestones of this project

For that aim, the work has been planned in the following milestones:

1. Documentation and study of existing NLP technologies, comparatives.
    a. Information search, comparative between technologies

b. Review articles and specific documentation of each
c. Management of the RTVE API, familiarization with HTTP requests
d. Development in Node JS

2. NER system development

a. Evaluation of R system
b. Evaluation of Java system
c. Comparative between R and Java systems
d. Evaluation of LUIS system (Microsoft Azure)
e. Evaluation of NLU system (IBM Watson)
f. Selection of best tools, debug code

3. TM system development

a. Review major NLP methods
b. Familiarization with LDA and its input and output variables
c. Specific pre-processing of documents
d. Obtaining vocabulary
e. Study of obtained topics
f. Re-adjustment of parameters

4. Combination of NER and TM tools

a. Joint system, correct incompatibility
b. Final results

5. Study of the results, assessment, documentation of the process

a. Evaluation measures
b. Evaluate results
c. Document all the work done, memory
d. Reviews

For these developments, existing NLP systems have been used, which are explained in later sections, and as a database, documents from the RTVE API.

B.2 Natural Language Processing

The automatic processing of natural language has an interpretive purpose, that of obtaining information implicit in the context or in other elements that do not appear in the discourse. It is a field of computing, that has been developed since 1950, when the first automatic translators were designed and guidelines from the field of “generative linguistics” studied, with descriptions based on rules of syntactic structures. These systems followed a series of rules defined and structured to face the problems to solve.

From that time and until 1960, United States develop an Automatic Translation Project, that laid the foundations for all subsequent development. As in all areas of computing, at the beginning language processing was developed based on a list of instructions

implemented manually in the operating code. However, since the end of the 1980s and until the mid-1990s, the "statistical revolution" has taken place, changing the approach that until now was given to the NLP (through rules and instructions), to resolve it also through Machine Learning. Since then, all NLP techniques have improved considerably.

NLP techniques are classified according to the information they wish to characterize, as shown in Table 2.1. Main technologies existing in this area are presented and compared in Table 2.2.

Named Entity Recognition

The recognition of named entities is a sub-task of Information Extraction, which seeks to locate and classify the appearances of an entity named in unstructured text in predefined categories, such as the names of people, organizations, places, codes doctors, expressions of time, amounts, monetary values, percentages, etc.

For the recognition of the most common entities there are already trained models, but the technologies allow to retrain models to identify the entities that interest us. This is especially useful when existing models do not have support for identifying entities in a certain language.

There are three types of learning:

- Rules-based models (conventional approach, instructions)
- Models based on labels (supervised learning, in which during the learning the model is readjusted comparing the results obtained with the solutions that should be obtained)
- Unsupervised models

In this project NER techniques are used with models based on labels (supervised learning). The process to train a NER model in a supervised manner is the following:

1. A set of documents is prepared as long as possible, correctly labelled with the type of entities to be located.
2. This set of documents is used to train the new entity recognition model. The system, opaque for the user, applies mathematical functions to infer a pattern or characteristics that the system finds in common in all the entities that it wishes to locate. Typically, it is suggested that there is heterogeneity among the set of training samples, so that the system is capable of managing various cases, as well as a sufficient number of samples. For a sufficient number of examples this system usually provides good results.
3. Finally, the list of entities obtained can be viewed as a list, or, according to the tool, in a more visual environment, which then allows using these entities for further processing.

Topic Modeling

Topic Modeling techniques belong to unsupervised models, since there is no "template" with the correct solutions to compare during learning. The model draws conclusions, and the designer evaluates them. If you think they are right or good, keep the model. If it is considered that the results can be improved, by means of changes in the implemented system, other situations can be tested and the results evaluated for each case.

In the field of natural language processing, a topic model is a statistical model that tries to identify or detect the implicit patterns that characterize the themes or common areas of the words that appear in a text, in an unsupervised way, as has explained. The system consists of three phases:

1. During the **training** of the system a "**corpus**" of a large number of documents is provided with some relation to each other, for example, having a similar structure, dealing with the same subject or being the same type of documents in terms of length. The system processes the documents and analyses in which documents the words usually appear, and in which usually the same ones co-occur.

   Groups of words that tend to explain similar documents will form a "topic" or common theme. That topic is characterized as a probabilistic mixture of words: some terms will appear with very high probabilities within the topic, while others will have almost zero probability because they are not characteristic for that topic.

   The model created during the training explains the patterns found in the corpus documents, and provides a list of the topics learned, as well as the distribution of topics of the training documents (probabilities of each topic to appear within each document).

2. During the training, the system has characterized the documents of the corpus through a **list of topics** (formed by some terms and their probabilities of appearing) and the **proportion of each topic in the documents**. Using this model of already created topics, it is possible to obtain a prediction of the composition of topics in a test document, that is, a new document not used during the training.

3. The results obtained can be **visualized** in some graphic environment, or used for other subsequent processing. The model draws conclusions in two aspects:
   - The distribution of topics in the documents
   - The distribution of the characteristic terms within a topic.

B.3 Design of the technical solution

The development of the system has been carried out by using diverse technologies (Table 3.1, and scheme in Fig. 3.1). As explained, the sample data has been obtained from the RTVE API.

NER module

The first task in NER work block was the comparison between different technologies. For this, R and Java were selected, and a NER system was developed in both of them:

1. **Handling of the text and entities**. First, pre-processing of the text files to eliminate the codes that may remain, and to prevent errors in the subsequent processing. Annotation of words and phrases in the text. Then, tag the people, places and organizations entities, using the pre-trained NER model. Extraction of those entities: for each document that has been processed, a file is created with the list of localized entities, and in turn another file is created with the text this time marked with the labels.

2. At this point, the **results** can be **evaluated**: the entities found by the system are compared with the ones that actually exist, in order to obtain error metrics from each of the systems. In order to compare the results obtained with the real solutions, all the entities of interest in the text are labelled manually. In other situations, there may be files previously labelled with the entities, but when working with the RTVE files these templates are not available with the solution. It could be possible also to look for another system that performs this process with more agility and good results, but to ensure that the solutions have a reliability as reliable as possible, we proceed to do it manually.

3. Obtained both labelled files (the result of automatic processing and labelling manually), **compare** the number and characteristics of the **entities correctly labelled** and those that do not. Also, by observing the entities labelled incorrectly or not labelled, patterns or situations that have led the machine to act in that way can be perceived. For example, words with a capital letter in the middle of a sentence, or sets of words labelled as a single entity when there are two.

TM module

Within the part of the mode of the topics that has been carried out in the first place a task of research and documentation of the current paradigm (see section 2.2.3 for more details).

Subsequently, follow the procedure of developing a TM system in Python, which, from 1,297 documents of about 100 sentences in length, the most recurrent terms in documents that respond to a theme or that make up a topic.

The development has followed the following stages:

1. Handling the text. Pre-processing of the documents that are going to be used, proceeding from RTVE's API. Clean the text and tokenize (divide into units or tokens, which will be the words). To identify the meanings of words, synonyms, antonyms, etc. In English the Wordnet library can be imported, which is a reference page of the English dictionary type. This library also allows

"lemmatizing" (table 2.1 of NLP subtasks), which consists of reducing the words or tokens to the most basic form of word that has meaning and appears in the dictionary. (It should not be confused with pruning or stemming, which reduces the word to its lexeme or root, although it has no meaning as a word in itself). However, because of being texts in Spanish, the lemmatization is done from the previously imported Spacy library, which separates in words and obtains the "lemma" and the grammatical function of each one. After this words' grammatical function, it filters the nouns that are significant in the text.

2. The text is ready to do the TM. Inside the directory, all the files in the folder are read, and a list of tokens or words present in each file is drawn up. Using the LDA model of the "corpora" module (plural of "corpus"), a dictionary is created from the list of words in the file. This dictionary is used to create the "corpus", the set of data that will be used to create the topics model.

3. The gensim module allows you to select the number of passes to the corpus to train the LDA model, and the number of topics that you wish to view. Now the appearance percentages of each topic within a document can be obtained, and seen what is the main topic and its probability. This information is also related to the terms (in English, "terms" that characterize each topic). From pyLDAvis we can visualize the list of words that characterize the topics, but we can also see a diagram (Fig. 3.3) of the interrelation of the topics with each other in the set of documents (on the left), and the terms within each topic (to the right).

Joint system

The interest of using these two modules is the integration in a joint system, for a more complete extraction of information in the text. The results of Named Entity Recognition system and Topic Modeling will be contrasted to relate the information retrieved from each of them.

1. Firstly, the text is pre-processed, so that the Topic Modeling system can handle the entity tags. For that aim, the punctuation marks are removed, and the labelling is changed from the R tag style (e.g., "<START:person>Federer</END>") to the Python formatting ("entidad_person_Federer").

2. Then the entities who were kept in the dictionary for TM are stored with their id. The dictionary creates a count of all the words that appear in the corpus of documents, and then filter the verbs, prepositions, adverbs or adjectives, along with the words that have very low probability of occurrence (they are so infrequent that they do not characterize the thematic) or too high. From these probabilities the most relevant topic or topics for a certain entity can be retrieved, and also the probabilities of occurrence from one and another topic in an entity.

3. A further comparison can be carried out by obtaining the similarity between the probability vectors of these entities. This way the entities that have a similar behaviour in terms of their appearances within a theme can be discovered.

B.4 Results and evaluation

NER

A NER system for identifying a type of entity is a problem of detection, or classification into two classes (decision). Each of the words in a text corresponds or not to an entity, and the decision maker must assign to each word the category "entity" (positive output) or the category "non-entity" (negative output).

Therefore, the success in the recognition of the entities of a text is a quantifiable number, which will relate the number of correctly labelled entities, with the number of those that were labelled without being entities, and entities that were not labelled.

Tabla B.1: In green, correctly classified words (entity / non-entity) and in red, incorrectly

Entities	Labelled	Not labelled
Real	True Positives: labelled entities	False Negative: not-labelled entities
False	False Positive: labelled not-entities	True Negative: not-labelled not-entities

The results of the NER process are compared in the next table for both systems, R and Java:

Tabla B.2: R (left) and Java (right) results on NER process, corresponding to Tabla 4.4 and Tabla 4.5 results.

Entities	Labelled	Not labelled	Total:
Real	~ 96	64 not found	160 entities
False	6	8.625 words	8.631 “no-entities”
Total:	102 labelled	8.689 ignored	8.791 words

Entities	Labelled	Not labelled	Total:
Real	~ 79	81 not found	160 entities
False	10	8.621 words	8.631 “no-entities”
Total:	89 labelled	8.702 ignored	8.791 words

Based on these cases, the error metrics can be obtained:

Tabla B.3: Error measures for both technologies. It can be observed that R exceed Java results.

Measure	Meaning	Expression	R	Java
Accuracy	percentage of correct answers on the total number of cases	$\frac{TP+TN}{TP+TN+FN+FP}$	99,20%	99,01%
Precision	Correct positive versus positive ones (false and true)	$\frac{TP}{TP+FP}$	94,12%	88,76%
Recall, Sensitivity	positive hits on what should have been (detected and not detected). Also "True Positive Rate"	$\frac{TP}{TP+FN}$	60%	49,37%
Specifity	"non-entities" correctly not labelled on what should have been. Also "True Negative Rate"	$\frac{TN}{TN+FP}$	99,93%	99,88%
F1 score	Harmonic mean of the results	$2\frac{Precision * Recall}{Precision + Recall}$	73,28%	63,45%

TM

The case of TM is completely different. As it is an unsupervised technology, there is no unique solution, nor the perfect solution, nor any recognizable patterns to be identified. The assignment of topics to a text can be very varied without ceasing to be correct, and there is no quantitative method for assessing the results in a standardized way.

Without a measure of error or success of the topics comes into play the user's criteria to accept them as reasonable or discard them. The most computationally calculated parameter to measure the convergence of the topics learning is the **perplexity**, which characterizes the evolution in the training of a topics model until it is stabilized, to find the optimal point in which to stop training. It is obtained from (B.1) formula, presenting Fig. B.2 results in this project.

$$perplexity\ (D) = \exp(-\frac{\sum_{d=1}^{M} \log p\ (w_d)}{\sum_{d=1}^{M} N_d}) \qquad (B.1)$$

Where D is the corpus in which it is evaluated, M is the number of documents, p (w_d) likelihood of the document, approximated through the distributions learned, and N_d the number of words of each document [12]. This parameter is obtained from the estimation of likelihood, according to the formula (B.2). The likelihood exact value cannot be obtained, but is approached through the upper bound.

$$perplexity = 2^{-\ln(Likelihood)} \qquad (B.2)$$

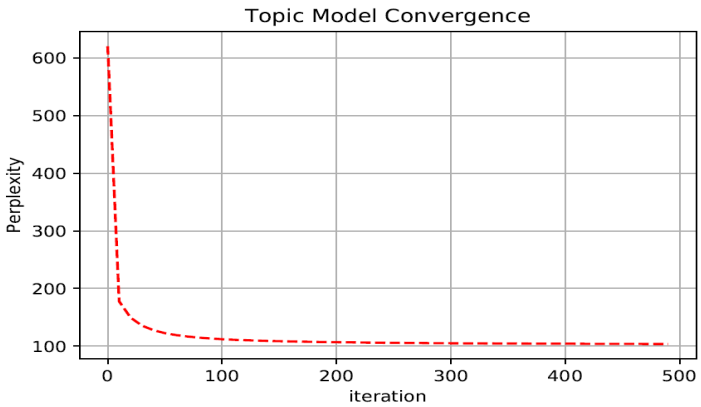


Fig. B.1: Convergence graph using perplexity in the trained Topic Model.

However, these parameters (perplexity, log-likelihood) are not sufficient in themselves to characterize the quality of the topics obtained. It is shown in chapter 4 (Fig 4.1) the comparison between two examples of perplexity curves, and how, despite the difference between the values of one and the other, having reached their convergence, they are humanly perceived as equally good. Other systems evaluate the quality of the models through **Human Task** studies [34], in which a common test consists of randomly replacing a term within a topic, and requesting to identify the term intruder. For example:

Topic 1 {“cat”, “sand”, “radio”, “clean”, “sofa”, “lamp”}

would be substituted by:

*Topic 1** {“cat”, “sand”, “radio”, “Toledo”, “sofa”, “lamp”}

Sometimes, these intruders are easy to identify, but in other cases where the terms are not so related, it may happen that the intruder does not stand out against the other terms. The more related the terms are to each other, the more information they provide and the better the topic will be. This test is known as an **intrusion test**.

Parallel to the studies with subjects there are investigations trying to obtain coherence metrics that provide results similar to those that a human being would give, although even the best results do not reach 80% accuracy on human responses. Therefore, a new family of measures is proposed to evaluate the quality of these topics’ models: **coherence** [35], which is applied to the "n" main words within a topic. It is typically defined as the average of the measures of similarity of the words in a document, obtained by pairs (each word with all the others). Those models that have more related topics will have higher coherence measures.

Tabla B.4: Some of the coherence results for this topic model, sorted by coherence values. Full table can be found as Tabla 4.7

Nº	**Coherence (npmi)**	**Terms in topic**
1	0.0025630357621744915	guerra, personaje, alemán, final, película, historia, ajedrez, puerta, momento, verdad
2	0.0004802325361501071	tiempo, planeta, energía, ciudad, mundo, tierra, edificio, especie, temperatura, aspecto
3	-0.008555883666803358	tierra, bosque, animal, hembra, neandertal, especie, preso, entidad_location_europa, costa, territorio
…	…	…
13	-0.1316515746291053	película, entidad_person_segunda, entidad_location_españa, tiempo, director, rodaje, actor, noche, momento, entidad_person_john
14	-0.16435155330147563	entidad_person_enrique, esposo, catalina, orden, caballero, hombre, reunión, entidad_person_ana, entidad_location_europa, silbato
15	-0.4320333282652562	célula, castaño, entidad_organization_adn, harina, molécula, envejecimiento, combinación, entidad_location_córcega, pizarra, creps

The coherence results obtained are quite low and very similar from one to another, due to the training for the coherence was in an external corpus, and because these systems are optimized for English topics and terms, in English documents.

Topic modelling seeks to obtain the main themes that appear in the documents. Performing the reverse search, for a topic among existing ones, you get the document in which that topic is the majority. The theme that groups these words is shown as an example:

['barco', 'edificio', 'fruto', 'metro', 'madre', 'hembra', 'hormigón', 'cabaña', 'tiempo', 'piedra']

For which the document in which it has greater importance that topic belongs to a documentary of the series "Great Designs", on constructions, that effectively, deals with a subject related to those terms:

> *(...) sí. pero la nueva perspectiva de entidad_person_Fred y entidad_person_Saffron durará poco. un repentino frío anuncia el comienzo del invierno, y el último vertido de hormigón para los techos se ha retrasado casi un mes. en diciembre, la familia tuvo que mudarse con su nueva casera ahora mismo están mi hijo, su mujer y sus dos hijos viviendo conmigo. ahora los míos. ya están, sí. ya los he limpiado. los limpió anoche. (...) un trabajo que se suponía que tardaría tres meses, ha tardado cerca de siete. sí, ha sido un trabajo difícil. el acceso ha sido complicado. hemos revisado el programa, pero la calidad no se mide con la rapidez. ya en el año siguiente, el granero del piso superior ya está, y la primera pieza de madera se abre paso. este debería ser un día trascendental, pero entidad_person_Saffron tuvo malas noticias de parte de los contratistas. (...) no pude hacer mucho. la dificultad de entidad_person_Saffron viene de la mano de un duro invierno. pero en abril, ella y entidad_person_Fred han reestructurado sus finanzas. (...)*

Also, the graphic results of the trained topics model are shown above, by using the visualization with pyLDAvis library.

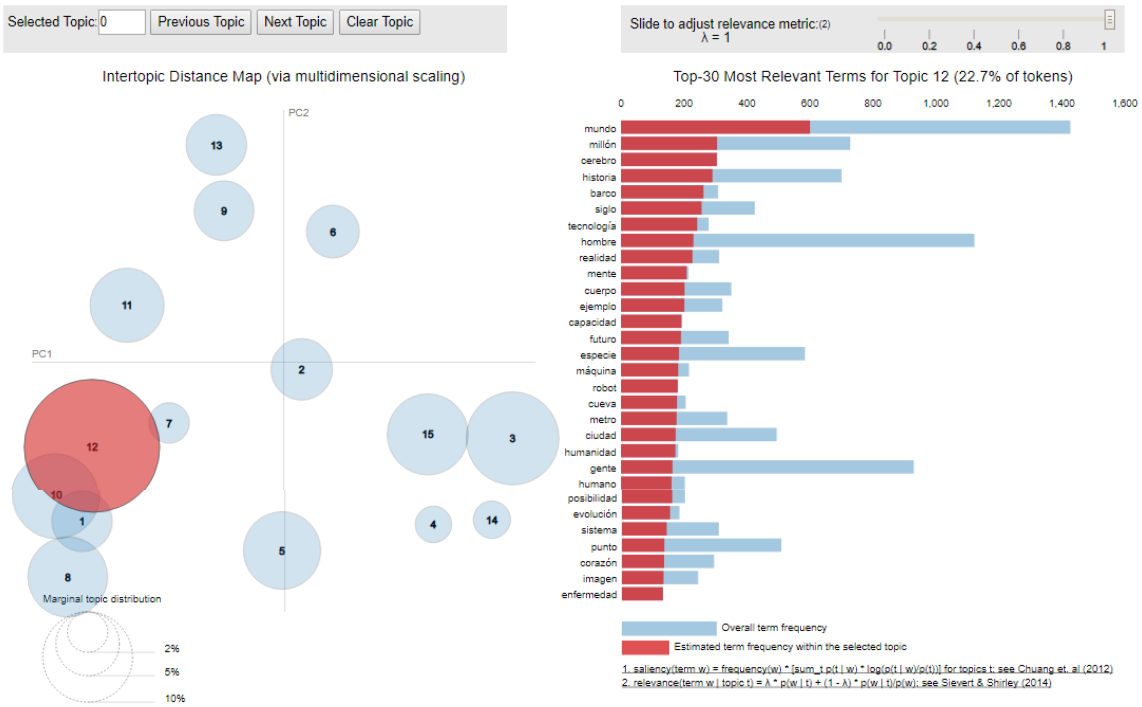


Fig. B.2: On the left, bubbles that represent the topics obtained. Topic 12 has been selected (in red), so that on the right the probabilities of the 30 majority terms within the topic appear for that document (red), as opposed to the general average (blue)

Joint system

The joint text processing system that has been developed consists of a NER module and another TM, whose results complement each other to provide more complete knowledge on the information contained in the text.

To evaluate the relationship between the results of both processes, the similarity metric called "cosine similarity" is used, over a vector of probabilities of every entity appearing in each topic This measurement is based on the scalar product between two vectors, which are projected on a single direction through the cosine of the angle they form. Thus, two vectors in the same direction will have a maximum scalar product and equal to the product value of their modules, while two vectors perpendicular to each other will form an angle of 90º, whose cosine is zero, and will have no similarity because they are perpendicular. Mathematically, being A_i and B_i components of A and B vectors:

$$Cosine\ similarity = \cos(\theta) = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2} \cdot \sqrt{\sum_{i=1}^{n} {B_i}^2}} \qquad \text{(B.3)}$$

Tabla B.5: For the entity of type place "Europe", the distribution of importance in appearances in the themes appears in the first row. Just below are the 10 entities with an appearance distribution in the most similar topics. Full table can be found as Tabla 4.9.

	Terms that characterize the 15 *topics* obtained for this documents' corpus	millón, continente, gracia, entidad_location_australia, desierto, tierra, historia, mundo, fruto, túnel	entidad_person_enrique, esposo, catalina, orden, caballero, hombre, reunión, entidad_person_ana, entidad_location_europa, silbato	hombre, gracia, señor, verdad, músico, tiempo, padre, entidad_organization_ríe, favor, entidad_person_ah	…	célula, castaño, entidad_organization_adn, harina, molécula, envejecimiento, combinación, entidad_location_córcega, pizarra, creps	tierra, bosque, animal, hembra, neandertal, especie, preso, entidad_location_europa, costa, territorio	entidad_person_el, entidad_location_madrid, campeón, derecho, semana, final, españa, jugador, momento, victoria	tiempo, planeta, energía, ciudad, mundo, tierra, edificio, especie, temperatura, aspecto	…	película, entidad_person_segunda, entidad_location_españa, tiempo, director, rodaje, actor, noche, momento, entidad_person_john	jenny, entidad_person_vincent, músico, reginald, chico, mundo, gracia, entidad_person_ah, entidad_location_barcelona, noche	…
	Topic id	**0**	**1**	**2**	**…**	**6**	**7**	**8**	**9**	**…**	**12**	**13**	**…**
0	entidad_ location_ europa	0.	0.47478195	0.	…	0.	0.49350444	0.	0.02400231	…	0.	0.	…
	The closest entities after relevance for these topics are the following:												
1	entidad_ person_ república	0.	0.	0.	…	0.	0.97490989	0.	0.0259011	…	0.	0.	…
2	entidad_ person_ cárpatos	0.	0.	0.	…	0.	1.	0.	0.	…	0.	0.	…
3	entidad_ person_ urss	0.	0.	0.	…	0.	1.	0.	0.	…	0.	0.	…
	…	…	…	…	…	…	…	…	…	…	…	…	…
9	entidad_location_ santa	0.	0.	0.	…	0.	1.	0.	0.	…	0.	0.	…
10	entidad_location_ eslovaquia	0.	0.	0.	…	0.	1.	0.	0.	…	0.	0.	…
11	entidad_organization_naciones	0.	0.	0.	…	0.	0.90874709	0.	0.	…	0.	0.	…
12	entidad_location_ puente	0.	0.	0.	…	0.	0.83108	0.	0.16892	…	0.	0.	…
13	entidad_location_ austria	0.	0.	0.	…	0.	0.79321047	0.	0.	…	0.20678953	0.	…
14	entidad_person_antoine	0.	0.96863203	0.	…	0.	0.	0.	0.	…	0.	0.	…

B.5 Conclusions

During this project, a study of the actual paradigm of Natural Language Processing has been carried out. Firstly, there has been a study of the existing systems and current technologies that are being developed and investigated, as part of the state of art. A historical review of the first technologies and their development up to the present moment has been made, presenting the current technologies and investigation areas (Tables 2.1 and following).

Next, the project has been divided in two branches of NLP, that will later converge. In the first of them, Named Entity Recognition was explored, through the implementation of a person, organizations and locations entity extractors. The subtitling documents of RTVE were used for this aim. Four systems were chosen to compare the benefits of each one, two of them, based on open source languages and libraries, and two of them, proprietary software. The phases of development in R and Java technologies through Apache OpenNLP library (Table 3.1 for the whole list of used tools) were completed, with satisfactory results (presented in section 4: Table 4.3 and following), although the other two technologies that were evaluated, LUIS by Azure Microsoft, and NLU by IBM Watson, weren't fully implemented. The stage of development that was reached in each of them is discussed in section 3.4.

The second branch was Topic Modeling for the same set of documents. After a brief review of existing mechanisms, LDA modelling was chosen. From the corpus of documents a LDA model is trained using Python's NLTK and Gensim libraries (full table of used tools in Table 3.1). This model creates a dictionary of significant words that appear in the corpus, and makes the count of appearances. Using this information, 15 topics are retrieved, characterized by a list of significant terms. The quality of these models is not quantifiable, but through different measures explored (section 4.2.2) the results are found satisfactory.

The purpose of these two systems is to complement each other for the extraction of more complete information from textual data. Both systems' results are contrasted (section 4.2.3), with results that met the desired objectives.

There are several open lines for future work, discussed in chapter 6.2, that would perfect the functioning of the developed system, so that it might be used in RTVE's audiovisual content platforms (on demand TV system "Televisión a la carta", mobile app and smart TV). It would be useful for identifying the content in the -subtitled- videos and documents, so that they might be automatically labelled in categories and classified for enhancing files retrieval of certain theme or entity. Also, this will enhance user experience and will allow new recommendations after the previously accessed content.